CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. (2015)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3449

RESEARCH ARTICLE

A systematic literature review of Linked Data-based recommender systems

Cristhian Figueroa[1,2,*,†], Iacopo Vagliano[1],
Oscar Rodríguez Rocha[1] and Maurizio Morisio[1]

[1] *Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Turin, Italy*
[2] *Universidad del Cauca, Calle 5 No. 4–70, Popayán, Colombia*

SUMMARY

Recommender systems (RS) are software tools that use analytic technologies to suggest different items of interest to an end user. Linked Data is a set of best practices for publishing and connecting structured data on the Web. This paper presents a systematic literature review to summarize the state of the art in RS that use structured data published as Linked Data for providing recommendations of items from diverse domains. It considers the most relevant research problems addressed and classifies RS according to how Linked Data have been used to provide recommendations. Furthermore, it analyzes contributions, limitations, application domains, evaluation techniques, and directions proposed for future research. We found that there are still many open challenges with regard to RS based on Linked Data in order to be efficient for real applications. The main ones are personalization of recommendations, use of more datasets considering the heterogeneity introduced, creation of new hybrid RS for adding information, definition of more advanced similarity measures that take into account the large amount of data in Linked Data datasets, and implementation of testbeds to study evaluation techniques and to assess the accuracy scalability and computational complexity of RS.

Received 4 August 2014; Revised 27 October 2014; Accepted 15 November 2014

KEY WORDS: Linked Data; recommender systems; systematic review; web of data

1. INTRODUCTION

The increasing amount of heterogeneous information available on the Web has led to the difficulty in recommending relevant items that meet the requirements of end users. It has attracted the attention of researchers and has become an interesting research area from the development of the first *recommender systems* (RS) in the mid-1990s [1–3]. In fact, the interest in this area remains high because of the abundance of practical applications that help users to deal with different kinds of information [4].

Nowadays, RS are increasingly common in many application domains, as they use analytic technologies to suggest different items or topics that can be interesting to an end user. However, one of the biggest challenges in these systems is to generate recommendations from the large amount of heterogeneous data that can be extracted from the items. Accordingly, some RS have evolved to exploit the knowledge associated to the relationships between data of items and data obtained from different existing sources [5]. This evolution has been possible, thanks to the rise of the Web

*Correspondence to: Cristhian Figueroa, Politecnico di Torino, Corso Duca degli Abruzzi, 24 - 10129, Turin, Italy.
†E-mail: cristhian.figueroa@polito.it

supported by a set of best practices for publishing and connecting structured data on the Web known as *Linked Data* [6].

Linked Data principles have lead to semantically interlink and connect different resources at data level regardless of the structure, authoring, location, and so on. Data published on the Web using Linked Data have resulted in a global data space called the Web of Data. Moreover, thanks to the efforts of the scientific community and the W3C Linked Open Data (LOD) project[‡], more and more data have been published on the Web of Data, helping its growth and evolution.

This work summarizes the state of the art of RS that make use of the structured data published as Linked Data on the Web. We undertook a systematic literature review, which is a form of secondary study that uses a well-defined methodology to identify, analyze, and interpret all available evidences related to specific research questions in a way that is unbiased and (to a degree) repeatable [7, 8]. We considered the most relevant problems that RS intended to solve, the way in which studies addressed these problems using Linked Data, their contributions, application domains, and evaluation techniques that they applied to assess their recommendations. Analyzing these aspects, we deduced current limitations and possible directions of future research. Unlike other works reporting the state of the art in RS [4, 9–11], our systematic literature review is the first to study RS that obtain information from Linked Data in order to generate recommendations.

The remainder of this paper is structured as follows. Section 2 provides a background information about Linked Data and RS. Section 3 summarizes the methodology and defines objectives and research questions. Section 4 outlines the results of the review organized according to each research question defined in Section 3. Section 5 discusses the results as well as the limitations of our systematic literature review. Section 6 contains the conclusions and future work. Finally, we list the selected papers in Appendix A.

2. BACKGROUND

2.1. Linked Data

In 1994, Tim Berners-Lee[§] uncovered the need of introducing semantics into the Web to extend its capabilities and to publish structured data on it, which became known as *Semantic Web*. The set of good practices or principles for publishing and linking structured data on the Web is known as Linked Data. While the Semantic Web is the goal, Linked Data provides the means to make it a reality [6]. The set of Linked Data principles are as follows:

- Use URI (uniform resource identifiers) as names for things.
- Use HTTP (Hypertext Transfer Protocol) URIs, so that people can look up those names.
- Use of standard mechanisms to provide useful information when someone looks up a URI, for example, RDF (Resource Description Framework) to represent data as graphs and SPARQL (SPARQL Protocol and RDF Query Language) to query Linked Data.
- Include links to other URIs, so that they can discover more things.

The main benefit of using Linked Data as a source for generating recommendations is the large amount of available concepts and the relationships between them that can be used to infer relations more effectively in comparison to derive the same kind of relationships from text [12]. As Linked Data information is machine-readable, it is possible to query datasets on a fine-grained level in order to collect information without having to take manual actions; therefore, information is explicitly represented, which allows for applying reasoning techniques when querying datasets and making implicit knowledge explicit.

2.2. Recommender systems

RS are software tools and techniques that provide suggestions of items to a user. These items can belong to different categories or types, for example, songs, places, news, books, films, and events.

[‡]http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
[§]http://www.w3.org/Talks/WWW94Tim

DOI: 10.1002/cpe

According to Adomavicius and Tuzhilin [4], the roots of RS can be traced back to the works in cognitive science, approximation theory, information retrieval, forecasting theories, management science, and consumer choice modeling in marketing.

Nowadays, RS are focused on the recommendation problem of guiding users in a personalized way to interesting items in a large space of possible options [10]. Typically, RS are classified as content based, collaborative filtering, knowledge based, and hybrid [5].

Content-based RS make suggestions that take into account the ratings that users give to items according to their preferences and the content of the items (e.g., extracted keywords, title, pixels, and disk space) [10]. Collaborative-filtering RS generate recommendations of items to a user taking into account ratings that users with similar preferences have given to these items [13]. Knowledge-based RS infer and analyze similarities between user requirements and features of items described in a knowledge base that models users and items according to a specific application domain [14]. Hybrid RS combine one or more of the aforementioned techniques in order to improve recommendations.

With the evolution of the Web toward a global space of connected and structured data, a new kind of knowledge-based RS has emerged known as Linked Data-based RS. This kind of RS suggests items taking into account the knowledge of datasets published under the Linked Data principles. The systematic literature review presented in this paper is focused on this kind of RS.

3. RESEARCH METHODOLOGY

This work studies the state of the art in Linked Data-based RS. It follows the guidelines set out by Kitchenham and Charters [8] for systematic literature reviews in software engineering. These guidelines provide a verifiable method of summarizing existing approaches as well as identifying challenges and future directions in the current research. Figure 1 presents the protocol for our systematic literature review.

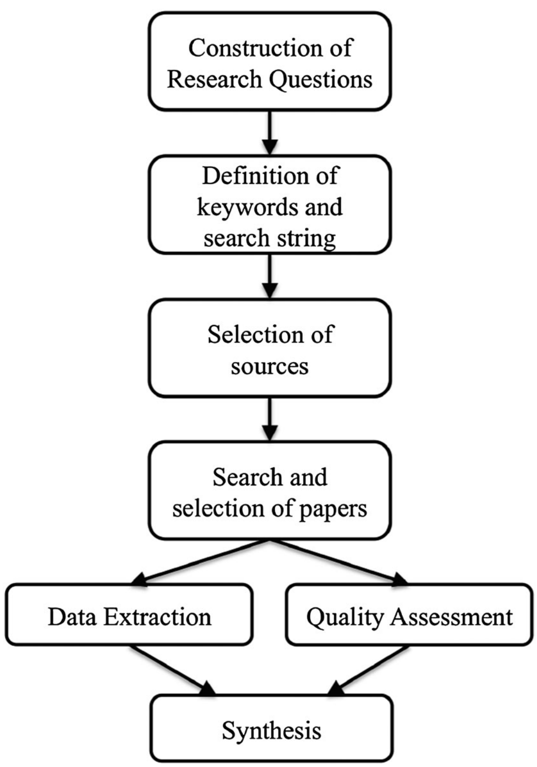


Figure 1. Systematic literature review at a glance.

DOI: 10.1002/cpe

The protocol is defined in order to setup the steps to conduct the systematic literature review. In our work, it was developed by the first and second authors, while the third and fourth authors validated it.

3.1. Construction of research questions, definition of keywords and search string, and selection of sources

The goal of our systematic literature review is to understand how the implicit knowledge, stored in Linked Data datasets and represented as concepts and relations between them, can be exploited to make recommendations. Accordingly, we have defined the following research questions:

RQ1 What studies present RS based on Linked Data?
RQ2 What challenges and problems have been faced by researchers in this area?
RQ3 What contributions have already been proposed (e.g., algorithms, frameworks, and engines)?
RQ4 How is Linked Data used to provide recommendations?
RQ5 What application domains have been considered?
RQ6 What criteria and techniques are used for evaluation?
RQ7 Which directions are the most promising for future research?

Afterwards, a preliminary set of keywords was defined: {*Linked Data, Recommender system*}. This set was then extended by searching for synonyms in order to obtain the final set of keywords used to define a search string. The search string is the query to look for papers in a set of online digital libraries. In this work, the search string that we defined is as follows:

```
("semantic web" OR "linked data" OR "web of data" OR "linked
open data") AND (recommendation OR "recommender system" OR
"recommendation system" OR "semantic recommendation" OR "semantic
recommender").
```

Furthermore, we selected seven scientific digital libraries that represent primary sources for computer science research publications as can be seen in Table I. Other sources like DBLP, CiteSeer, and Google Scholar were not considered as they mainly index data from the primary sources.

3.2. Search and selection

The studies selected in this systematic literature review were identified from the selected sources during March 2014. In Table II, a set of inclusion/exclusion criteria were defined in order to determine whether or not a study should be included.

3.3. Quality assessment, data extraction, and synthesis

We have defined a set of quality criteria that are listed in the checklist provided in Table III. Quality for each question is typically scored with values 1, 0.5, and 0, in order to represent the answers *'yes'*, *'partly'*, and *'no'*.

First and second authors evaluated the selected studies using this checklist. To do this, the total set of selected papers was split into two disjoint subsets, and each author selected only one of these

Table I. Sources selected for the search process.

Source	URL
IEEExplore	http://ieeexplore.ieee.org
SpringerLink	http://link.springer.com
Scopus	http://www.scopus.com
ACM Digital Library	http://dl.acm.org
Science Direct	http://www.sciencedirect.com
ISI Web of Knowledge	http://apps.webofknowledge.com
Wiley Online Library	http://onlinelibrary.wiley.com

DOI: 10.1002/cpe

Table II. Inclusion and exclusion criteria.

Inclusion criteria
Papers presenting recommender systems (RS) using Linked Data to provide recommendations.
Papers addressing exploratory search systems using Linked Data. Exploratory search refers to cognitive consuming search such as learning or topic investigation. Exploratory search systems also recommend relevant topics or concepts, although the key difference with respect to RS is that they still require an input query (commonly a set of keywords).
Papers from conferences and journals.
Papers published from 2004 to 2014. Linked Data is a relative new technology; therefore, RS approaches exploiting it are also recent.
Only papers written in English language.
Short and workshop papers that fulfill the above criteria: we had no reason to believe that they would fail to provide sufficient levels of detail about their studies.
Exclusion criteria
Papers not addressing RS neither exploratory search systems.
Papers addressing RS or exploratory search systems that do not exploit Linked Data to produce recommendations.
Papers addressing similarity measures but not RS. Similarity is a broader topic than RS.
Papers that use Semantic Web techniques (e.g., rule-based or ontology-based reasoning) but not Linked Data.
Papers that report only abstracts or slides of presentations because of the lack of information.
Grey literature. We do not think that technical reports, unpublished studies, and PhD thesis would add much more information with respect to journal and conference papers.

subsets to evaluate the papers. After this evaluation, cross-checking of the assessment was done on arbitrary studies (about 30 % of selected papers) by the third author. Finally, an agreement on differences was reached by discussion.

Data extraction was done in parallel with the quality assessment. We split the set of included studies into two disjoint subsets. First and second authors performed the task on a subset, then the third author cross-checked a random sample of 30% of studies. The data extracted are presented in Table IV.

The synthesis step is based on the methodology for thematic synthesis described by Cruzes and Dybå [15]. This methodology defines codes as descriptive labels applied to segments of text from each study. We defined an initial set of codes based on research questions, and subsequently, we performed a second coding with more precise codes, which were closer to the content of selected papers. The coding was performed by first and second authors: each of them addressed a subset of the papers as for data extraction and quality assessment, because it was done in parallel with them. Then, the third author performed again the coding on a random sample of 30% of papers for cross-checking; afterwards, disagreements were solved by discussion.

4. RESULTS

This section summarizes the relevant information found in the selected studies in order to answer the proposed research questions. A further discussion and analysis of these results are addressed in Section 5.

4.1. Included studies

RQ1 regards the studies that present RS based on Linked Data. We retrieved 69 papers to include in the systematic literature review, corresponding to 52 unique primary studies (a study is a unique research work that can include one or more papers). These studies were published in conferences, workshops, and journals between 2004 and 2014. The criteria for deciding the most significant

DOI: 10.1002/cpe

Table III. Quality assessment checklist.

Question	Score
Q1. Did the study clearly describe the challenges and problems that is addressing?	yes / partly / no (1 / 0.5 / 0)
Q2. Did the study review the related work for the problem?	yes / partly / no (1 / 0.5 / 0)
Q3. Did the study discuss related issues and compare with the alternatives?	yes / partly / no (1 / 0.5 / 0)
Q4. Did the study recommend the further continuous research?	yes / partly / no (1 / 0.5 / 0)
Did the study describe the components or architecture of the proposed recommender system?	yes / partly / no (1 / 0.5 / 0)
Q5. Did the study describe the components or architecture of the proposed recommender system?	yes / partly / no (1 / 0.5 / 0)
Q6. Did the study provide empirical results?	– The study provided an implementation of its work with an empirical evaluation and it was used in real applications, e.g., by other services (1)
	– The study provided an implementation of its work and an empirical evaluation but was not referred or used in other studies/applications (0.75)
	– The study provided an implementation only (0.5)
	– The study did not provide any implementation but it was referred by other works as a base on which start (0.25)
	– The study did not provide any implementation and was not referred by other works (0)
Q7. Did the study provides a clear description of the context in which the research was carried out?	yes / partly / no (1 / 0.5 / 0)
Q8. Did the study presents a clear statement of findings?	yes / partly / no (1 / 0.5 / 0)

paper for each study were completeness and publication year. The final set of selected papers and corresponding studies can be found in Appendix A.

With regard to the quality assessment, *journals* and *conference* studies have better quality than *workshop* studies as shown in Figure 2. Conference studies have the biggest spread, while journal studies, the lowest. In any case, the quality score is higher than 0.5 for all paper types, that is, rather good according to the quality criteria defined in Section 3.3.

4.2. *Research problems*

In order to address RQ2, we summarize the main problems involved in the studies considered and regarding the production of accurate recommendations. Table V lists these problems according to the number of studies in which they occurred. The number of studies represents the occurrence of each problem in the selected studies, which may be addressed in more than one study. The same applies for the rest of the results reported in this section.

In the following, we describe each item of Table V:

Lack of semantic information It was the most frequent problem in the selected studies, and it concerns the need for exploiting the rich semantics of information about items. Possible causes of this problem are as follows:

- Data about items are unstructured.
- A categorization of the items is needed.

DOI: 10.1002/cpe

Table IV. Data extraction form.

Data field	Description	Research question
ID	—	—
Title	—	—
Authors	—	—
Year of publication	—	—
Year of conference	—	—
Volume	—	—
Issue	—	—
Location	—	—
Proceeding title	—	—
ISBN	—	—
Publisher	—	—
Examiner	Name of person who performed data extraction	—
Publication source	—	—
Context	Environment in which study was conducted: industry, academic, government	—
Population	Study participants: students, academics, practitioners, etc.	—
Aims	Goals of the study (in our opinion when not clearly reported by authors)	—
Research problem	—	RQ2
Application domain	—	RQ5
Contributions	—	RQ3
Criteria and techniques for evaluation	—	RQ6
Findings	—	—
Limitations	—	RQ7
Future work	—	RQ7
Notes	—	—
Other information	—	—

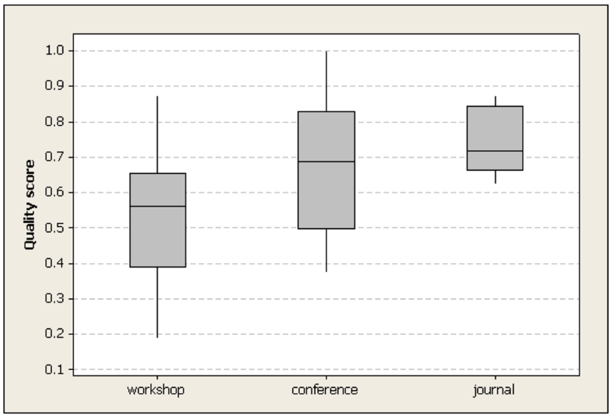


Figure 2. Quality score for different types of study.

- It is necessary to find relationships to link items.
- Social information is lacking.
- It is necessary to acquire content-descriptive metadata.
- Similarity measures that take into account semantic information are needed.

Complexity of information about items It is related to the complexity of information because of noisy metadata about features of items. Other causes for this problem are semantic heterogeneity and distribution of resources. The latter can impact on maintenance of the knowledge bases and can also decrease the accuracy of recommendations.

DOI: 10.1002/cpe

Table V. Distribution of studies according to the problems they addressed.

Problems	Number of studies
Lack of semantic information	13
Complexity of information about items	12
User dependency	8
Cold-start	6
Data quality	6
Computational complexity	5
Data sparsity	5
Domain dependency or specific and limited domain	4
Other problems	2

Table VI. Distribution of studies according to the contributions provided.

Contribution	Number of study
Algorithms	27
Similarity measures	12
Ontologies	8
Information aggregation or enrichment	8
Others	16

User dependency In a number of cases, RS require users to perform manual operations to acquire information about their profiles and interests. Such operations can be user feedback, ratings, filtering, attaching content-descriptive metadata, and semantic annotation of items.

Cold-start It is a well-known problem found mainly on RS based on collaborative-filtering approaches. Cold-start is a situation in which there are not enough ratings for items in order to generate recommendations.

Data quality This problem occurs when the knowledge base used to acquire information for providing recommendations is not reliable. Problems affecting data quality can range from poor reliability (e.g., wrong links between concepts or incorrect representations) to poor quality of recommended items.

Computational complexity It is related to the high computational demand that RS require to produce recommendations because of the large amount of data about items.

Data sparsity This is related to the lack of information about users or items and generates low density of significant data or connections.

Domain dependency It occurs when recommendations are only useful for items in a specific and limited domain without taking into account data that can be obtained from other related domains.

Other problems They include the need for recommending relevant and yet unknown items and the overspecialization of RS.

4.3. Contributions

In order to address RQ3, we classified the contributions provided by each study. Table VI shows the different kind of contributions and the number of studies in which they occurred (each study possibly reports more than one contribution).

The two main contributions are the definition or extension of a similarity measure and the definition or extension of an ontology, accounting for 12 and eight studies respectively. Algorithms are also addressed by 27 studies in total. Finally, information aggregation or enrichment and various other contributions account for eight and 16 studies, respectively. In the following, we describe each item of Table VI:

Algorithms Most of the selected studies proposed new algorithms or extensions of algorithms existing in the literature. In particular, four categories emerged: defining of a new algorithm,

DOI: 10.1002/cpe

adapting an algorithm to Linked Data, combining of algorithms to obtain a new hybrid algorithm, and extending of an existing algorithm. The definition of a new algorithm was the most frequent in 15 studies, while the adaptation of an algorithm to Linked Data, the combination of algorithms to obtain a new hybrid algorithm, and the extension to an algorithm each account for 4 studies. Furthermore, we can group algorithms into two classes:

- Graph-based algorithms, which compute relevance scores for items represented as nodes in a graph. A number of algorithms in this category are (*i*) the weight spreading activation algorithm, which propagates the initial score of a source node through its weighted edges; (*ii*) algorithms that update the scores of its linked nodes; (*iii*) algorithms that explore concepts and relations defined in an RDF graph; (*iv*) topic-based algorithms, which find similar items belonging to the same categories of an initial concept; and (*v*) path-based algorithms to find semantic paths between documents in the RDF graph.
- Algorithms to produce recommendations based on statistical information techniques applied to Linked Data such as support vector machine (SVM), latent Dirichlet allocation (LDA), random indexing (RI), and scaling methods. SVM analyzes and recognizes patterns in RDF triples; LDA is based on the co-occurrence of terms; RI uses distributional statistics to generate high-dimensional vector spaces; and scaling methods take into account the probability that an item could be selected based on its popularity (the number of entities is directly connected with the node). In addition, some algorithms define item-user matrices to compute semantic similarity based on path-lengths.

Similarity measures The selected studies applied a variety of similarity measures. These include pairwise cosine function for vector similarity computation between items, feature-based similarity to evaluate semantic distance on different datasets, rating-based similarity to compute the popularity of items among users, semantic relatedness defined by vocabulary meta-descriptions, content similarity that exploits lexical features, expressivity closeness based on the language constructs adopted, distributional relatedness derived from vocabulary usage, and topic-based similarity that captures the relatedness between items based on the categories they belong to.

Ontologies A number of studies proposed ontologies to assist or improve the recommendation process. New ontologies were proposed to facilitate the process of integration of datasets from a number of domains in order to make RS more flexible to changes, while a combination of existing ontologies described different types of entities such as users and items. Furthermore, it was found that reusing existing ontologies or vocabularies enable interoperability. Ontologies are also used to represent semantic distances, their explanations, user preferences, and item contents. A number of ontologies that are used in selected studies for these purposes are FOAF (Friend Of A Friend), SIOC (Semantically-Interlinked Online Communities), Resource List Ontology and Bibliographic ontology.

Information aggregation or enrichment This refers to the contributions about the aggregation of data to item collections and enrichment of existing ontologies or vocabularies. This is useful, for example, to obtain descriptive information about items and find entities in datasets in order to infer links between them. One contribution of this type is the aggregation of information from a specific domain when items have to be enriched with knowledge contained only on specialized datasets, another is the enrichment databases of RS with shared vocabularies.

Others Other contributions include the integration of other techniques such as opinion aggregators, exploitation of trust in web-based social networks to create predictive RS, and the use of social-based algorithms to improve the performance of the RS.

4.4. Use of Linked Data

Another interesting aspect that we studied was the use of Linked Data in RS, as underlined by RQ4. We classified the selected studies according to the way they used Linked Data to produce recommendations and grouped them into the following:

Linked Data driven RS that rely on the knowledge of the Linked Data to provide recommendations. For example, RS that calculate a semantic similarity based on diverse relationships that can

DOI: 10.1002/cpe

Table VII. Distribution of studies according to the use of Linked Data.

Category	Number of studies
Linked Data driven	37
Hybrid	29
Hybrid and Linked Data driven	21
Linked Data driven only	13
Representation only	10
Hybrid only	6
Exploratory search	4
Exploratory search and Linked Data driven	4
Exploratory search only	0

be found between concepts of Linked Data datasets and are related to features or descriptions of items. Such relationships can be paths, links, or shared topics among a set of items. This category can also include RS that use other techniques applied on data obtained from Linked Data datasets, for example, weight spreading activation, vector space model (VSM), SVM, LDA, and random indexing.

Hybrid RS that exploit Linked Data to perform some operations that can be used or not used to provide recommendations. This means that hybrid RS include Linked Data driven RS, which use recommendation techniques that rely on Linked Data, and RS that use Linked Data in other operations (not necessarily for recommending) that can be preliminary to the recommendation process (e.g., to aggregate more information from other datasets, to describe user profiles, or to annotate raw data in order to extract information to be integrated and used for recommending).

Representation only RS in this category exploit the RDF format to represent data and use at least one vocabulary or ontology to express the underlying semantics. However, no information is extracted from other dataset, and Linked Data are not used to provide recommendations. An example is an RS that represents the information about the users according to FOAF vocabulary but does not exploit Linked Data for other operations.

Exploratory search These systems are not RS, but their main duty is to assist users to explore knowledge and to suggest relevant to a topic or concept. Exploratory search systems and RS use Linked Data in a very similar way, although the key difference is that exploratory search systems still require an explicit input query (commonly a set of keywords). Additionally, users in these systems are not only interested in finding items but also in learning, discovering, and understanding novel knowledge on complex or unknown topics [16].

Each study may be assigned to more than one category; that is, it can be both Linked Data driven and hybrid, or both exploratory search and Linked Data driven. The only exception is for the representation-only category, in which studies cannot belong to other categories.

Table VII shows that most of the studies considered are Linked Data driven, and roughly 60% of them are also hybrid. Only 20% of hybrid studies were hybrid only, while the rest are also Linked Data driven. Moreover, 10 studies are representation only and just four exploratory search systems were included in the systematic literature review. All of the exploratory search studies are also Linked Data driven. This finding is consistent with the focus of the systematic literature review, which is on RS using Linked Data. It is worth noting that exploratory search is a broader topic; in this paper, we only consider the exploratory systems that recommend concepts to users.

The two most interesting categories are Linked Data driven and hybrid. Figure 3 shows the different techniques used by the studies in the first category to provide recommendations. The majority of them rely on datasets or on a similarity measure (about 43% and 35%, respectively), while the remaining 22% adapt natural language processing or content-based techniques or exploit reasoning.

Instead, Figure 4 illustrates the techniques that hybrid studies use together with Linked Data to provide recommendations. Most of them are natural language processing or collaborative-filtering methods (accounting for slightly less than 40% and about 35%, respectively) , and also reasoning or social networks are exploited in some cases.

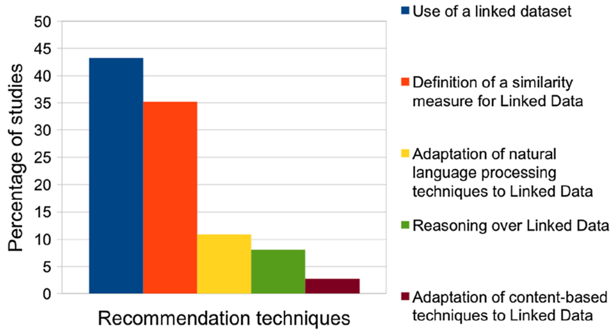


Figure 3. Distribution of Linked Data driven studies according to the recommendation techniques that they exploit (percentages refer to the total number of Linked Data-driven studies).

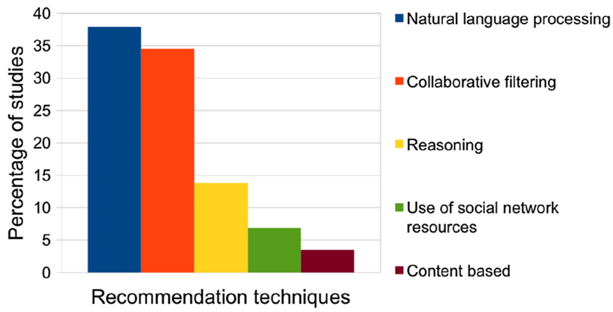


Figure 4. Distribution of hybrid studies according to the recommendation techniques that they exploit (percentages refer to the total number of hybrid studies).

Table VIII. Distribution of studies according to the Linked Data (LD) datasets on which they rely.

	Number of studies				
Dataset	General	LD driven	Hybrid	Hybrid and LD driven	LD driven only
DBpedia	31	28	20	16	12
Freebase	6	6	5	5	1
YAGO	4	3	3	2	1
Wordnet	4	2	3	2	0
DBLP	3	3	3	3	0
Dataset independent	3	3	3	3	0
LinkedMDB	3	3	3	3	0
Geonames	2	1	2	1	0
MusicBrainz	2	1	2	1	0
mySpace	2	2	2	2	0
ACM	1	1	1	1	0
IEEE	1	1	1	1	0
Eventseer2RDF	1	1	1	1	0
LinkedUp	1	1	0	0	1
mEducator	1	1	0	0	1
LinkedGeoData	1	0	1	0	0
LODE	1	1	1	1	0

In addition, we studied which datasets are used and the outcome is presented in Table VIII. It shows how many studies use a dataset overall and also considers the study category. It is possible to notice that DBpedia is used much more than the others. In fact, it is the biggest dataset, and it is the most curated.

DOI: 10.1002/cpe

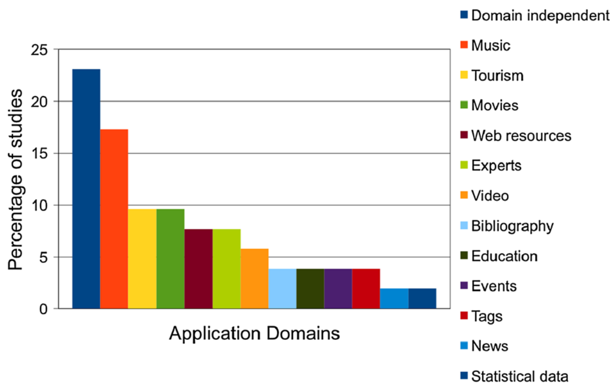


Figure 5. Distribution of studies according to the application domain.

Furthermore, it contains information about many different domains. Other commonly used datasets are Freebase, YAGO, and Wordnet, but the latter is used in just half of the cases by Linked Data-driven studies. In fact, it is also used with natural language processing techniques. On the contrary, the other datasets are used in most cases by Linked Data driven studies and often by studies which are both Linked Data driven and hybrid.

4.5. *Application domains*

Figure 5 illustrates the application domains considered by the studies selected for the systematic literature review. Most of the studies (about 23%) are not limited to any particular domain and can be used to recommend different kinds of items. Instead, an often occurring domain is music, which represents 17% and is followed by tourism and movies, accounting for roughly 10% each. Then there are web resources, expert recommendations, and video, with between 5% and 7% each, and a number of other domains are considered by the remaining 10% of the studies.

4.6. *Evaluation techniques*

RQ6 concerns RS evaluation, so we also dealt with this aspect. It is important to note that we focus on RS evaluation; thus, GUI evaluation is not considered, although some of the studies addressed it. RS are commonly evaluated according to their computational complexity and accuracy [17]. The former measures the execution time required to produce recommendations, which depends on the complexity of the algorithms used as well as the runtime of third-party systems needed to produce recommendations. The latter is the capacity of the RS to satisfy the individual user's need for information, and it can be evaluated by means of two techniques: user studies and comparison with similar methods. In this subsection, we detail both of them.

User studies involve users in order to compare recommendations generated by RS with the users' judgements or ratings. In these techniques, the most frequent measures are the following:

- Precision and recall, which evaluate the relevance of an RS taking into account the number of retrieved items, the number of items that evaluators considered as relevant, and the total number of available items.
- User ratings, which are techniques in which a list with results from different RS are presented to users who rate the lists according to their personal criteria [17].
- Ranking quality, which takes into account the retrieval correctness. The latter assigns an output ranking, a performance score based upon the available reference relevance judgments [18]. Common metrics to measure the ranking quality are the normalized discounted cumulated gain, average position, and presence.
- Unexpectedness of a concept suggestion, which is the degree of novelty of a recommendation for the evaluator.

DOI: 10.1002/cpe

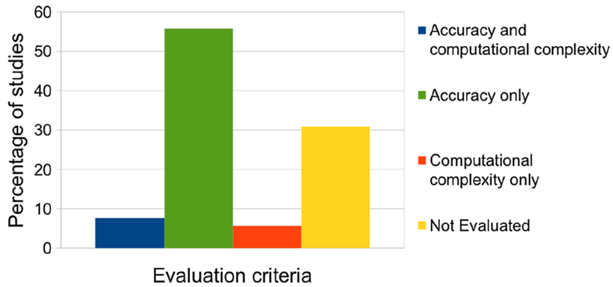


Figure 6. Distribution of studies according to the evaluation criteria (percentages refer to total number of studies).

Table IX. Distribution of studies according to the evaluation techniques.

Type	Technique	Number of papers
User studies	Precision and recall	18
	User ratings	9
	Ranking quality	3
	Unexpectedness	3
Comparison with similar methods	Precision and recall	5
	MAE and RMSE	3
Computational complexity	Execution time	7

MAE, mean absolute error; RMSE, root mean squared error.

In the case of comparisons with similar methods, recommendations generated by a specific RS are compared with well-known similar approaches. In the following, we mention the two main types:

- Precision and recall are measured, but in this case, items recommended by a well-known approach are considered as relevant.
- Mean absolute error (MAE) and root mean squared error (RMSE) are metrics to measure the predictive accuracy of an RS. MAE calculates the average absolute deviation between predicted similarities and similarity values in the real data set, while RMSE pays more attention to large errors [19].

Figure 6 shows the main evaluation techniques found in the selected studies, as well as their classification and their occurrence in these studies. Studies that provided an evaluation accounted for about 70% of the studies included in the systematic literature review. Among these, roughly 55% only used an accuracy technique, while roughly 2% only evaluated the computational complexity, and slightly less than 8% considered both accuracy and computational complexity.

Table IX details the techniques used in the studies included by considering the two types of accuracy evaluation and also computational complexity. The most frequent technique used to evaluate RS is the relevance measured with recall and precision metrics (used by 18 works in user studies and by about five in comparison with similar methods). We expected this result because these metrics are the ones most commonly deployed in information retrieval approaches. Other widely used techniques are user ratings, accounting for nine studies, and execution time, which is exploited by seven studies.

4.7. Future work

RQ7 is related to directions for future research. To address this, we summarized the future work that the selected studies proposed in order to extend or improve their approaches. Specifically, about 67% of studies included in the systematic literature review present diverse proposals for future work. Table X lists the most important, indicating for each one the number of studies in which it was mentioned. A deeper analysis of these results and a discussion of possible directions is presented in Section 5.

In the following, we provide a brief description of each item reported in Table X:

DOI: 10.1002/cpe

Table X. Distribution of studies according to the future work they propose.

Future work	Number of studies
Personalization of recommendations	8
Use more datasets	8
Create hybrid recommender systems	7
Similarity measures	4
Find more semantic relationships (item-user and item-item)	3
Other proposal for future work	3
Consider other domains	2

Personalization of recommendations The idea is to know to what extent personalization can improve recommendations without requiring user profile information or user intervention for manual operations (feedback, filtering, annotation, etc.).

Use more datasets It means to increase the range of data to annotate or match items to be recommended. It can also be useful to explore new domains because of the use of other datasets which can be from diverse domains.

Create hybrid RS This refer to exploring new ways to combine diverse recommendation techniques for creating hybrid approaches and improving the relevancy and quality of recommendations.

Similarity measures It is the creation of new similarity measures or the improvement of existing ones.

Find more semantic relationships It is the possibility of finding more semantic relationships between items and between users and items. It is considered by three studies.

Consider other domains Although domain dependency is one of the problems found in various studies, only two studies took into account exploring new application domains for providing recommendations.

Other proposal for future work This group includes applications in real life contexts, algorithms for categorization of recommendations, improving performance of algorithms, and the study of disambiguation techniques.

4.8. Limitations

The limitations reported in the selected studies are also related to RQ7 as these can help us to uncover the open issues in RS based on Linked Data and their relationships with proposals of future work. They are grouped into four main types: datasets, manual operations, personalization, and computational complexity. We detail each of them in the following:

Datasets This type describes limitations of RS due to the datasets used.

- A number of studies required a local copy of the entire dataset in a local server in order to reduce the runtime to produce recommendations. This had to be done as sometimes public datasets offer limited results, restricted access, and high timeout.
- Sometimes data had to be manually curated because of the poor reliability of public datasets.
- A number of RS are limited to the use of only one dataset. This can restrict the knowledge to which the RS can have access, avoiding data from diverse sources and domains being obtained.

Manual Operations It means that RS needed the user to perform manual operations in order to produce recommendations. Among these operations, we found:

- RS requiring manual selection of relevant concepts according to a specific application domain or interests. This is a difficult and tedious task considering the large amount of data that a typical Linked Data dataset can contain.
- RS that did not rank their results, so final users are faced with no priority in the recommendation.

DOI: 10.1002/cpe

Personalization It is about producing recommendations according to the user profile or some personal features.

Computational complexity RS still need to improve the performance because of high computational demand to analyze large amounts of items and information stored into datasets. Another problem is the poor performance of public endpoints to access them.

5. DISCUSSION

In the first part of this section, we present a discussion of the results considering each research question, while in the second part, we mention the limitations of our systematic literature review.

5.1. Specific research questions

This subsection discusses the research questions addressed in this systematic literature review according to the results reported in Section 4.

RQ1 is a general question regarding the studies that describe RS based on Linked Data. To provide an answer, we have followed the steps described in the protocol presented in Section 3 in order to search and select studies in this area. Firstly, we retrieved a total number of 7873 papers (including those duplicated) from scientific digital libraries. After each author filtered papers by title and abstract, we discussed disagreements, and we reach consensus on a final set of 69 papers to include in our study, which correspond to 52 unique studies.

RQ2 deals with research problems in the RS domain that researchers intended to solve by proposing approaches based on Linked Data. We found that the lack of semantic information and its complexity were the most notorious problems in RS.

Lack of semantics regards the need for rich semantic information about items. This is the main reason to devise novel strategies to represent items and user profiles using diverse semantic techniques exploiting several knowledge sources from the Linked Data cloud.

The complexity and heterogeneity of information and the subsequent cost of maintenance of knowledge bases make Linked Data a suitable solution that uses publicly available knowledge bases that are continuously growing and maintained by third parties. However, this poses new challenges, for example, the need for mechanisms to assure the reliability of these knowledge bases that are used to describe user profiles and items and to generate recommendations.

Domain dependency is another problem that has been also addressed by using Linked Data because it allows the possibility to exploit information from different datasets that can be domain-independent or belong to diverse domains. In fact this is one reason why the most used dataset is DBpedia as it is the most generic dataset that can be used for cross-domain RS. Nonetheless, some studies still report this problem as future work.

Computational complexity is a question that has not been widely addressed in the studies considered in this systematic literature review and remains as an open issue because most of the studies have concentrated only on semantic enrichment of items and inclusion of datasets in Linked Data cloud. Computational complexity needs to be addressed more because in RS not only accuracy is important but also scalability and responsiveness. For example, this problem can be critical in RS for mobile scenarios where users demand fast response times.

Other problems such as usability, cold-start, data quality, and data sparsity have been addressed by combining with Linked Data various techniques based on natural language processing, reasoning or social network resources, and creating hybrid RS that exploit both collaborative filtering and content-based approaches.

RQ3 inquires about the contributions proposed in RS based on Linked Data. The analysis showed that the majority of studies are focused not only on providing new algorithms but also on defining or extending a similarity measure of an ontology. Furthermore, adaptation, combination, or extension to algorithms is quite often addressed together with information aggregation

DOI: 10.1002/cpe

or enrichment. Accordingly, we found that Linked Data can be used in RS for several purposes such as the following:

- Defining different similarity functions between items or users by exploiting the large data available in the Linked Data cloud and the vast relationships already established such as properties or context-based categories. In this way, it is possible to extract semantic information from textual descriptions or other textual properties about the items in order to find semantic similarities based on the information stored in interlinked vocabularies of Linked Data. This can be useful in RS based on collaborative filtering to improve the neighborhood formation in user-to-user or item-to-item.
- Generating serendipitous recommendations, for example, to recommend items that are not part of the users' personal data cloud, that is, suggest new, possibly unknown items, to the user; or to guide users in the process of the exploration of the search space giving the possibility for serendipitous discovery of unknown information (for exploratory search systems).
- Offering the explanation of the recommendations given to the users by following the linked-data paths among the recommended items. In this way, users can understand the relationship between the recommended items and why these items were recommended.
- Domain-independency when creating RS as it is possible to access data from Linked Data datasets from different domains.
- Enrichment of information sources such as databases, repositories, and registries with information obtained from dataset in Linked Data cloud which manage huge amounts of data. It offers the possibility to enrich graphs representing users and/or items with new properties in order to improve graph-based recommendation algorithms. Additionally, it helps to mitigate the new-user, new-item, and sparsity problems.
- Annotating items and users with information from multiple sources facilitate RS to suggest items from different sources without changing their inner recommendation algorithms. Using such a semantic-based knowledge representation, recommendation algorithms can be designed independently from the domain of discourse.
- Obtaining hierarchical representation of items because the topic distribution that some datasets in Linked Data cloud offer. In this way, RS can base their recommendation on the exploration of items belonging to similar categories.

RQ4 regards the diverse ways in which Linked Data is used to provide recommendations. First of all, we classified the studies according to the way they exploited Linked Data. As reported in Section 4, four categories were identified: *Linked Data driven RS* rely mainly on Linked Data to perform their tasks, *hybrid RS* use Linked Data and also other techniques, *representation-only RS* do not provide Linked Data-based recommendations but use Linked Data for representing data based on RDF, and finally *exploratory search systems* that are not RS but may help users to find concepts or topics and have some similar features to RS especially in the use of Linked Data.

Table XI describes each category including the most important studies that adopted these strategies, as well as their advantages and disadvantages. The numbers of the studies corresponds to the identifiers in Appendix A.

Most of the studies belong to the first category, and many belong to both the first and the second category. These two categories are also the most interesting as they include RS to better exploit the advantages provided by Linked Data in order to reach best results. We also studied techniques to provide recommendations relying on Linked Data and slightly less than half of Linked Data driven RS used a dataset, almost one third define a similarity measure for Linked Data, while others adapt natural language processing or content-based methods or use reasoning.

With reference to the techniques used together with Linked Data, we found that natural language processing and collaborative filtering are the most used (both account for about one

DOI: 10.1002/cpe

Table XI. Classification of Linked Data-based RS approaches.

Approach	Techniques	Advantages	Disadvantages
Linked Data driven	– *Graph based*: weight spreading activation (S17), semantic exploration in an RDF graph (S29, S10, S3, S9, S19), and projections (S23) – *Reasoning*: (S1, S51) – *Statistical*: Matrix item-user (S29, S35, S31, S13, S37, S10), Scaling methods (S29) and topic discovery (S2)	– Generating serendipitous recommendations – Offering explanations of the recommendations following the linked-data paths – Creating domain-independent RS – Exploiting hierarchical information about items to categorize recommendations	– High cost of exploiting semantic features due to inconsistency of LD datasets – No personalization – No contextual information – High computational complexity – Need for manual operation – Need for dataset customization to address the computational complexity
Hybrid	– *Collaborative Filtering and Linked Data*: (S2, S4, S12, S25, S27, S3, S28, S26, S30, S35) – *Information aggregation and Linked Data*: opinions (S16), ratings (S19), and social tags (S32) – *Statistical methods and Linked Data*: Random Indexing (S10), VSM (S47, S31, S35), LDA (S35), Implicit feedback (S25), SVM (S13), Structure-based statistical semantics (S37)	– Overcoming the data sparsity problem – Allowing collaborative filtering RS to address the cold start problem	– High computational complexity
Representation only	– Item/user information representation using RDF-based ontologies (S36, S38, S20, S40, S14, S15, S42, S46)	– Improving scalability and reusability of ontologies – Easing data integration – Enabling complex queries	– Difficult to reuse the already available knowledge in the Linked Data Cloud
Explorative search	– Set nodes and associated lists (S49, S39, S34) – Spreading activation to typed graphs and graph sampling technique (S11)	– Enabling self-explanation of the recommendations	– No automation of the recommendation because explorative search approaches require frequent interaction with the user

DOI: 10.1002/cpe

third of hybrid RS) as they intended to provide personalized suggestions of items tailored to the preferences of individual users.

Other techniques are less common (less than 15%), and they are reasoning, use of social network resources, and content-based methods. Reasoning has not been widely used as its quality is still insufficient, and its coverage is not broad enough at the level of system components and knowledge elements [20]. Therefore, one solution is to develop RS based on reasoning-oriented natural language processing enriched with multilingual sources and able to support knowledge sources generated largely by people as Linked Data datasets.

As for the datasets used in the selected studies, we found that DBpedia is the most used Linked Data dataset. This is because DBpedia is a generic dataset and most of the studies are domain-independent that need to be evaluated in diverse scenarios. DBpedia is one of the biggest datasets that is frequently updated as it obtains data from Wikipedia that continuously grows into one of the central knowledge sources [21]. It makes Dbpedia multimodal and suitable for RS that need to be domain-independent and for knowledge-based RS where complexity and cost of maintenance of the knowledge base is high. However for RS of a single domain, it is better to use specific datasets but always implementing a linking interface with generic datasets in order to resolve ambiguities or to exploit unknown semantic relationships.

RQ5 concerns the application domains considered by RS based on Linked Data so far. We identified 12 domains, but we found that most of the RS are domain-independent (slightly more than one fifth of the studies). This is because most of the proposed recommendation algorithms can be applied in diverse domains by only changing the dataset or taking only a portion of it in order to obtain the data to generate the recommendations.

However, we also note that items of music, tourism, and movies are the most recommended as these belong to common domains in which there is a large amount of data and state-of-the-art datasets available, which allow the researchers to compare their results with several works developed in the community.

Accordingly, in a number of cases, the domain impacts also on datasets because they require a reduction of information; that is, only a subset of concepts is considered, which requires offline processing and more effort to maintain the dataset even if it improves the performance. For example, Passant developed RS named *dbrec* [22], which required to manually extract a subset of the data of DBpedia related with bands and musical artists.

RQ6 regards the evaluation techniques used to study RS based on Linked Data. We classified them into two types: accuracy and computational complexity. Accuracy evaluates recommendations according to their relevance for final users, while computational complexity measures the execution time required to produce them.

With regard to accuracy, our results demonstrate that researchers are more interested in evaluations made by final users than in comparisons with similar methods. This result was expected because usefulness of recommendations depends more on final user preferences than on comparing with similar approaches where evaluation may be biased as researchers must trust the results obtained. Therefore, future methodologies of evaluation should be user-centered in order to assure the quality of the results of RS.

Additionally as expected, most of the selected studies were more likely to evaluate their recommendations applying traditional methods of information retrieval such as precision and recall that are focused on percentages of true positives, false negatives, and false positives.

Interestingly, we found that few works evaluated the computational complexity of RS, which is a critical factor specially for applications that need responses with short timeouts. Therefore, it is still an open issue considering that accessing Linked Data datasets in most cases is time consuming and requires that researchers download dumps of the datasets to access them in local repositories.

RQ7 aimed to uncover the most promising directions for future research on RS based on Linked Data. To address this issue, we have reported not only future works but also limitations of the selected studies.

DOI: 10.1002/cpe

Section 4.7 summarized the future work reported in the selected studies. We found that the most frequently future works were the personalization of recommendations, the use of more datasets, and the creation of hybrid RS.

The lack of personalization of recommendations is still a common drawback in Linked Data-based RS. It concerns the fact that different users obtain the same set of results with the same input parameters. To solve this drawback, some RS need explicit feed back from users in order to differentiate the results based on information about the user's profile (e.g., browsing history and favorite music genre).

However, these approaches force the user to perform extra work like rating items or building an exhaustive user profiles. Consequently, there is a need of non-invasive personalization approaches supported by Linked Data in order to obtain implicit information from the neighborhood relationships user-to-user, item-to-item, and user-to-item. These relationships can be inferred from the links between concepts of datasets in Linked Data cloud related with properties of items and users.

Using more datasets is needed in order to increase the base of knowledge to produce recommendations. As presented in Section 4.8, there are some limitations of the current Linked Data-based RS with regard to the use of Linked Data datasets such as restricted access, poor reliability, computational complexity, low coverage of languages, domain dependency, and the need for installing a local copy of the dataset. For this reason, it is important to investigate new ways to integrate different datasets in order to (*i*) extend the knowledge base allowing the RS to access to other datasets in case that the main dataset fails or the data are not reliable; (*ii*) create scalable RS because they can be adapted to other domains by only accessing to the appropriate dataset and (*iii*) improve the performance by selecting datasets with better response time.

The creation of hybrid RS is not a new proposal, as could be seen in Section 4.4, combining diverse techniques of recommendation with Linked Data-based approaches is a frequent practice in the selected studies. However, we also found that it is still an open issue because it is necessary to investigate which combinations of techniques are more suitable for RS applied in diverse contexts. For example, combining Linked Data-based RS with social-based RS can be a good choice for applications that require information about the users and their interrelationships. In this way, RS can access information that sometimes is not available in Linked Data datasets such as items rating information, user profiles, and other social information.

The inclusion of user profile information (user profiling) is another aspect that is not widely considered in Linked Data recommender systems. The idea behind the user profiling is to obtain a meaningful concept-driven representation of user preferences in order to enable more precise specifications of user's preferences with less ambiguity. Therefore, this can be also useful to contribute to the personalization of Linked Data-based RS.

The automatic selection of the appropriate dataset according to the type of items or the application domain is another challenge that intend to improve the quality of recommendations. This dynamic process of selection can help the algorithms to choose the best strategy to find candidate items to be a recommender based on the implicit knowledge contained in Linked Data and the relationships with properties of items and users.

As a consequence, it is also important to study new similarity measures and techniques able to automatically combine information from different datasets and to deal with the diversity of data in these datasets. Furthermore, it can be possible to create a statistical models of user interests to overcome the topical diversity of rated items.

Finally, we found that there is still a need for building testbeds in order to allow for rigorous, transparent, and replicable testing and for studying new techniques (or adaptation of those existing) for evaluating the accuracy and computational complexity of RS based on Linked Data. This must also consider that Linked Data-based RS may have access to large amounts of information and that links among items can be unknown to the users. Additionally, large-scale RS should be also evaluated in terms of the ability to scale and provide recommendations with data coming from millions of users and/or items

DOI: 10.1002/cpe

5.2. *Limitations of our systematic literature review*

This section describes the main limitations we faced during our systematic literature review. Firstly, although some of selected papers were initially included because of their title or abstract, in the end they were excluded because we could not access them from our University.

Secondly, we only considered the most relevant paper for each study in order to calculate the frequency of problems, future work, contributions, and evaluation techniques. As a consequence, we could be biased, as some papers belonging to the same study may present a problem or contribution not reported in the most relevant paper.

Finally, we did not perform deep validation. Because of time issues, the majority of studies were read by one researcher, and cross-checking was performed only on about one third of the studies. Nonetheless, for some papers for which assessment was difficult, there was a discussion between the first three authors.

6. CONCLUSIONS

This systematic review has discussed 69 papers reporting 52 primary studies addressing RS that make use of the structured data published as Linked Data. We focused on identifying the most relevant problems that these studies aimed to solve and how they used Linked Data to provide recommendations. Although some of our results are already known, we defined a protocol to support our assumptions. Furthermore, we analyzed contributions, limitations, application domains, evaluation techniques they applied to assess their results, and the proposed directions for future research.

With regard to the research problems, we found that the most relevant ones were the lack of semantic information and the complexity of information about items. In order to overcome the lack of semantics, RS are enriched with diverse Linked Data datasets that are useful to describe users and items while reducing the ambiguity and exploiting the vast amount of links between related concepts stored in these datasets.

The majority of the selected studies have addressed these problems using Linked Data for several purposes, such as (*i*) finding new relationships or similarities based on links, paths, graphs, and created on the basis of Linked Data; (*ii*) generating serendipitous recommendations, that is, recommending items that are not expected by the users because of the links uncovered once the items are enriched with Linked Data; and (*iii*) explaining the recommendations, that is, allowing users to understand the reason of a recommendation by following the paths among items in the Linked Data cloud.

We also provided a classification of the selected studies according to the way they use Linked Data to provide recommendations. In particular, we identified four classes: Linked Data driven RS, which rely on techniques applied on datasets in Linked Data cloud such as categories, paths, number of input, and output links; hybrid RS that combine traditional techniques of recommendation (e.g., collaborative filtering and content based) with Linked Data; representation-only RS that uses Linked Data only to represent items or users but not for recommendations; and exploratory search systems that are not RS but help users to discover content through a guided search and are specially useful for users interested in learning or investigating a topic.

Additionally, we studied the most common datasets that RS use in order to obtain information, and we found that more than a half of these studies rely on DBpedia. This is because DBpedia is considered a central hub for the Linked Data cloud; it is linked to various datasets that gives the possibility to access diverse data from different application domains. Additionally, it makes DBpedia suitable for testing purposes in generic RS.

Concerning the evaluation techniques, the majority of the selected studies are focused on accuracy and rely more often on *user studies* than *comparison with other methods*. Computational complexity is also assessed in few cases; however, we think that it is an important factor to be evaluated especially for applications needing short responses such as RS in mobile environments. Additionally, we found that there is still a need for building testbeds to allow for testing and studying the results of RS based on Linked Data.

DOI: 10.1002/cpe

According to our findings, we identified that two recurrent issues in the selected studies are the high computational demand and the domain dependency. Therefore, we believe that further research is still needed to offer non-invasive personalization, exploit more datasets, and improve performance. Additionally, future work should focus on providing evaluation of RS considering the accuracy and computational complexity. With regard to application domains, music, movies, and tourism items are the most used in RS, and this may be due to the fact that in these domains, there are more datasets that help scientists to assess the results of their RS in comparison with similar approaches.

Finally, it is worth to mention that currently, we are working in the area of RS; in particular, we are developing RS that uses Linked Data as a source of information to recommend items for multiple application domains. The currently obtained results have been presented in [23], in which we describe how the RS based on Linked Data can be applied in the eTourism domain.

APPENDIX A. SELECTED PAPERS

Rows in italics identify papers (P) belonging to a study (S) already reported by other paper (e.g., papers 10, 19, and 54 belong to the same study S10).

Table A.1. Selected papers (P) and corresponding studies (S).

P	S	Authors	Year	Title	Publication details
1	S1	Fernández-Tobías, I., Cantador, I., Kaminskas, M., Ricci, F.	2011	A generic semantic-based framework for cross-domain recommendation	2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems - HetRec ’11, pp 25 - 32
2	S2	Kabutoya, Y., Sumi, R., Iwata, T., Uchiyama, T., Uchiyama, T.	2012	A Topic Model for Recommending Movies via Linked Open Data	International Conferences on Web Intelligence and Intelligent Agent Technology, pp 625–630
3	S3	Dell’Aglio, D., Celino, I., Cerizza, D.	2010	Anatomy of a Semantic Web-enabled Knowledge-based Recommender System	4th international workshop Semantic Matchmaking and Resource Retrieval in the Semantic Web, at the 9th International Semantic Web Conference, pp 115–130
4	S4	Mannens, E., Coppens, S., Wica, I., Dacquin, H., Van De Walle, R.	2013	Automatic News Recommendations via aggregated Profiling	Journal Multimedia Tools and Applications, 63 (2), pp 407–425
5	S5	Dzikowski, J., Kaczmarek, M.	2012	Challenges in Using Linked Data within a Social Web Recommendation Application to Semantically Annotate and Discover Venues	International Cross Domain Conference and Workshop, pp 360–374
6	S6	Wardhana, A.T.A.; Nugroho, H.T.	2013	Combining FOAF and Music Ontology for Music Concerts Recommendation on Facebook Application	Conference on New Media Studies, pp 1–5
7	S7	Passant, A., Raimond, Y.	2008	Combining Social Music and Semantic Web for music-related recommender systems	First Workshop on Social Data on the Web, pp 19–30
8	S8	Lindley, A., Graf, R.	2011	Computing Recommendations for Long Term Data Accessibility basing on Open Knowledge and Linked Data	5th ACM Conference on Recommender Systems, pp 51–58

DOI: 10.1002/cpe

Table A.1. *Continued.*

P	S	Authors	Year	Title	Publication details
9	S9	Passant, Alexandre	2010	dbrec–Music Recommendations Using DBpedia	The Semantic Web–ISWC 2010, pp 209–224
10	S10	Stankovic, M., Breitfuss, W., Laublet, P.	2011	Discovering Relevant Topics Using DBPedia: Providing Non-obvious Recommendations	2011 International Conferences on Web Intelligence and Intelligent Agent Technology, 1, pp 219–222
11	S11	Marie, N., Gandon, F., Ribière, M., Rodio, F.	2013	Discovery Hub : on-the-fly linked data exploratory	9th International Conference on Semantic Systems, pp 17–24 search
12	S12	Peska, L., Vojtas, P.	2013	Enhancing Recommender System with Linked Open Data	10th International Conference on Flexible Query Answering Systems, pp 483–494
13	S13	Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D.	2012	Exploiting the web of data in model-based recommender systems	6th ACM conference on Recommender systems
14	S14	Golbeck, J.	2006	Filmtrust: movie recommendations from semantic web-based social networks	3rd IEEE Consumer Communications and Networking Conference, pp 1314–1315
15	S15	Celma, Ò., Serra, X.	2008	FOAFing the music: Bridging the semantic gap in music recommendation	Web Semantics: Science, Services and Agents on the World Wide Web, 6 (4), 250–256
16	S16	Varga, B., Groza, A.	2011	Integrating DBpedia and SentiWordNet for a tourism recommender system	7th International Conference on Intelligent Computer Communication and Processing, pp 133–136
17	S17	Kaminskas, M., Fernández-Tobías, I., Ricci, F., Cantador, I.	2012	Knowledge-based music retrieval for places of interest	Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies–MIRUM '12, pp 19–24
18	S18	Dietze, S.	2012	Linked Data as facilitator & practice for TEL recommender systems in research	2nd Workshop on Recommender Systems for Technology Enhanced Learning, pp 7–10
19	*S10*	*Damljanovic, D., Stankovic, M., Laublet, P.*	*2012*	*Linked Data-Based Concept Recommendation : Comparison of Different Methods*	*9th Extended Semantic Web Conference, pp 24–38*
20	S19	Kitaya, K., Huang, H. H., Kawagoe, K.	2012	Music curator recommendations using linked data	Second International Conference on the Innovative Computing Technology, pp 337–339
21	S20	Jung, K., Hwang, M., Kong, H., Kim, P.	2005	RDF Triple Processing Methodology for the Recommendation System Using Personal Information	International Conference on Next Generation Web Services Practices, pp 241–246
22	S21	Calì, A., Capuzzi, S., Dimartino, M. M., Frosini, R.	2013	Recommendation of Text Tags in Social Applications Using Linked Data	ICWE 2013 Workshops
23	*S21*	*Calì, A., Capuzzi, S., Dimartino, M. M., Frosini, R.*	*2013*	*Recommendation of Text Tags Using Linked Data*	*3rd International Workshop on Semantic Search Over the Web, pp 1–3*
24	S22	Meymandpour, R., Davis, J. G.	2012	Recommendations using linked data	5th Ph.D. workshop on Information and knowledge–PIKM '12, pp 75–82
25	S23	Harispe, S., Ranwez, S., Janaqi, S., Montmain, J.	2013	Semantic Measures Based on RDF Projections: Application to Content-Based Recommendation Systems	On the Move to Meaningful Internet Systems: OTM 2013 Conferences SE–44, pp 606–615

DOI: 10.1002/cpe

Table A.1. *Continued.*

P	S	Authors	Year	Title	Publication details
26	S24	Hopfgartner, F., Jose, J. M.	2010	Semantic user profiling techniques for personalised multimedia recommendation	Multimedia Systems, 16 (4-5) pp 255–274
27	*S5*	*Łazaruk, S., Dzikowski, J., Kaczmarek, M., Abramowicz, W.*	*2012*	*Semantic Web Recommendation Application*	Federated Conference on Computer Science and Information Systems (FedCSIS), pp 1055–1062
28	S25	Ostuni, V. C., Di Noia, T., Di Sciascio, E., Mirizzi, R.	2013	Top-N recommendations from implicit feedback leveraging linked open data	Proceedings of the 7th ACM conference on Recommender systems, pp 85–92
29	S26	Ahn, J., Amatriain, X.	2010	Towards Fully Distributed and Privacy-Preserving Recommendations via Expert Collaborative Filtering and RESTful Linked Data	International Conference on Web Intelligence and Intelligent Agent Technology, pp 66–73
30	S27	Heitmann, B., Hayes, C.	2010	Using Linked Data to Build Open, Collaborative Recommender Systems	AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, pp 76–81
31	S28	Zarrinkalam, F., Kahani, M.	2012	A multi-criteria hybrid citation recommendation system based on linked data	2nd International eConference on Computer and Knowledge Engineering (ICCKE), 2012, pp 283–288
32	S29	Lommatzsch, A., Kille, B., Kim, J. W., Albayrak, S.	2013	An Adaptive Hybrid Movie Recommender based on Semantic Data	10th Conference on Open Research Areas in Information Retrieval, pp 217–218
33	S30	Torres, D., Skaf-Molli, H., Molli, P.; Díaz, A.	2013	BlueFinder: Recommending Wikipedia Links Using DBpedia Properties	5th Annual ACM Web Science Conference, pp 413–422
34	S31	Ostuni, V. C., Di Noia, T., Mirizzi, R., Romito, D., Di Sciascio, E.	2012	Cinemappy : a Context-aware Mobile App for Movie Recommendations boosted by DBpedia	International Workshop on Semantic Technologies meet Recommender Systems & Big Data SeRSy 2012, pp 37–48
35	S33	Zhang, Y., Wu, H., Sorathia, V., Prasanna, V. K.	2008	Event recommendation in social networks with linked data enablement	15th International Conference on Enterprise Information Systems, pp 371–379
36	S34	Mirizzi, R., Di Noia, T.	2010	From exploratory search to web search and back	3rd workshop on Ph.D. students in information and knowledge management–PIKM ’10, pp 39–46
37	S35	Khrouf, H., Troncy, R.	2013	Hybrid event recommendation using linked data and user diversity	Proceedings of the 7th ACM conference on Recommender systems, pp 185–192
38	S36	Bahls, D., Scherp, G., Tochtermann, K., Hasselbring, W.	2012	Towards a Recommender System for Statistical Research Data	2nd International Workshop on Semantic Digital Archives
39	S37	Cheng, Gong; Gong, Saisai; Qu, Yuzhong	2011	An Empirical Study of Vocabulary Relatedness and Its Application to Recommender Systems	10th International Conference on The Semantic Web – Volume Part I, pp 98–113
40	S38	Wang, Y., Stash, N., Aroyo, L., Gorgels, P., Rutledge, L., Schreiber, G.	2008	Recommendations based on semantically enriched museum collections	Web Semantics: Science, Services and Agents on the World Wide Web, 6 (4), 283–290
41	*S11*	*Marie, N., Gandon, F., Legrand, D., Ribière, M.*	*2013*	*Discovery Hub: a discovery engine on the top of DBpedia*	*3rd International* Conference on Web Intelligence, Mining and Semantics
42	S31	Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., Zanker, M.	2012	Linked open data to support content–based recommender systems	8th International Conference on Semantic Systems

Table A.1. *Continued.*

P	S	Authors	Year	Title	Publication details
43	*S31*	*Ostuni, Vito Claudio; Gentile, Giosia; Noia, Tommaso Di; Mirizzi, Roberto; Romito, Davide; Sciascio, Eugenio Di*	*2013*	*Mobile Movie Recommendations with Linked Data*	*International Cross–Domain Conference, pp 400–415*
44	*S31*	*Mirizzi, R., Di Noia, T., Ragone, A., Ostuni, V. C., Di Sciascio, E.*	*2012*	*Movie recommendation with DBpedia*	*3rd Italian Information Retrieval Workshop, pp 101–112*
45	S39	Waitelonis, J., Sack, H.	2011	Towards exploratory video search using linked data	Multimedia Tools and Applications, 59 (2), pp 645–672
46	S40	Li, S., Zhang, Y., Sun, H.	2010	Mashup FOAF for Video Recommendation LightWeight Prototype	7th Web Information Systems and Applications Conference, pp 190–193
47	S41	Hu, Y., Wang, Z., Wu, W., Guo, J., Zhang, M.	2010	Recommendation for Movies and Stars Using YAGO and IMDB	12th International Asia–Pacific Web Conference, pp 123–129
48	S42	Ruotsalo, T., Haav, K., Stoyanov, A., Roche, S., Fani, E., Deliai, R., Mäkelä, E., Kauppinen, T., Hyvönen, E.	2013	SMARTMUSEUM: A mobile recommender system for the Web of Data	Web Semantics: Science, Services and Agents on the World Wide Web, 20, pp 50–67
49	S43	Stankovic, M., Jovanovic, J., Laublet, P.	2011	Linked Data Metrics for Flexible Expert Search on the Open Web	8th Extended Semantic Web Conference, pp 108–123
50	S44	Ozdikis, O., Orhan, F., Danismaz, F.	2011	Ontology-based recommendation for points of interest retrieved from multiple data sources	International Workshop on Semantic Web Information Management, pp 1–6
51	S45	Debattista, J., Scerri, S., Rivera, I., Handschuh, S.	2012	Ontology-based rules for recommender systems	International Workshop on Semantic Technologies meet Recommender Systems & Big Data, pp 49–60
52	S46	Codina, V.; Ceccaroni, L.	2010	Taking Advantage of Semantics in Recommendation Systems	2010 Conference on Artificial Intelligence Research and Development, pp 163–172
53	*S9*	*Passant, A., Decker, S.*	*2010*	*Hey! Ho! LetŠs Go! Explanatory Music Recommendations with dbrec*	*7th Extended Semantic Web Conference, pp 411–415*
54	*S10*	*Stankovic, M.,* Breitfuss, W., Laublet, P.	*2011*	*Linked-data based* suggestion of relevant topics	*7th International* Conference on Semantic Systems, pp 49–55
55	*S9*	*Passant, A.*	*2010*	*Measuring semantic distance on linking data and using it for resources recommendations*	*AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, pp 93–98*
56	*S14*	*Golbeck, J.*	*2006*	*Generating Predictive Movie Recommendations from Trust in Social Network*	*4th International Conference, iTrust 2006, pp 93–104*
57	*S39*	*Sack, H.*	*2009*	*Augmenting Video Search with Linked Open Data*	*International Conference on Semantic Systems, pp 550–558*
58	S47	Baumann, S., Schirru, R., Streit, B.	2011	Towards a Storytelling Approach for Novel Artist Recommendations	8th International Workshop, AMR 2010, Linz, Austria, August 17–18, 2010, Revised Selected Papers, pp 1–15
59	S48	Corallo, A., Lorenzo, G., Solazzo, G.	2006	A Semantic Recommender Engine Enabling an eTourism Scenario	10th International Conference, pp 1092–1101

DOI: 10.1002/cpe

Table A.1. *Continued.*

P	S	Authors	Year	Title	Publication details
60	S49	Nuzzolese, A. G., Presutti, V., Gangemi, A., Musetti, A., Ciancarini, P.	2013	Aemoo: Exploring Knowledge on the Web	Proceedings of the 5th Annual ACM Web Science Conference, pp 272–275
61	*S49*	*Musetti, A., Nuzzolese, A., Draicchio, F., Presutti, V., Blomqvist, E., Gangemi, A., Ciancarini, P.*	*2012*	*Aemoo: Exploratory Search based on Knowledge Patterns over the Semantic Web*	*Semantic Web Challenge*
62	*S47*	*Baumann, S., Schirru, R.*	*2012*	*Using Linked Open Data for Novel Artist Recommendations*	*13th Internal Society for Music Information Retrieval Conference*
63	S50	Cantador, I., Castells, P.	2006	Multilayered Semantic Social Network Modeling by Ontology-Based User Profiles Clustering: Application to Collaborative Filtering	Proceedings of 15th International Conference, pp 334–349
64	*S34*	*Mirizzi, R., Ragone, A., Di Noia, T., Di Sciascio, E.*	*2010*	*Ranking the Linked Data: The Case of DBpedia*	*10th International Conference, pp 337–354*
65	S51	Heitmann, B., Hayes, C.	2010	Enabling Case-Based Reasoning on the Web of Data	The WebCBR Workshop on Reasoning from Experiences on the Web at International Conference on Case-Based Reasoning
66	S52	Alvaro, G., Ruiz, C., Córdoba, C., Carbone, F., Castagnone, M., Gómez-Pérez, J. M., Contreras, J.,	2011	miKrow : Semantic Intra-enterprise Micro-Knowledge Management System	8th Extended Semantic Web Conference, pp 154–168
67	*S50*	*Cantador, I., Castells, P., Bellogín, A.*	*2011*	*An Enhanced Semantic Layer for Hybrid Recommender Systems: Application to News Recommendation*	*Int. J. Semant. Web Inf. Syst., 7 (1), pp 44–78*
68	S32	Cantador, I., Konstas, I., Jose, J. M.	2011	Categorising social tags to improve folksonomy-based recommendations	Web Semantics: Science, Services and Agents on the World Wide Web, 9 (1), pp 1–15
69	*S29*	*Lommatzsch, A., Kille, B., Albayrak, S.*	*2013*	*A Framework for Learning and Analyzing Hybrid Recommenders based on Heterogeneous Semantic Data Categories and Subject Descriptors*	*10th Conference on Open Research Areas in Information Retrieval, pp 137–140*

REFERENCES

1. Hill W, Stead L, Rosenstein M, Furnas G. Recommending and evaluating choices in a virtual community of use. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95: ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995; 194–201.
2. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. GroupLens: an open architecture for collaborative filtering of netnews. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work—CSCW '94*, 1994; 175–186.
3. Shardanand U, Maes P. Social information filtering: algorithms for automating "word of mouth". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995; 210–217.
4. Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on* 2005; **17**(6):734–749.

DOI: 10.1002/cpe

5. Ricci F, Rokach L, Shapira B. Introduction to recommender systems handbook. In *Recommender Systems Handbook*. Springer, 2011.
6. Bizer C, Heath T, Berners-Lee T. Linked data—the story so far. *International Journal on Semantic Web and Information Systems* 2009; **5**(3):1–22.
7. Kitchenham B. Procedures for performing systematic reviews, Keele University: Eversleigh, Australia, 2004.
8. Kitchenham B, Charters S. Guidelines for performing systematic literature reviews in software engineering, University of Durham: Durham, UK, 2007.
9. Candillier L, Jack K, Fessant F, Meyer F. State-of-the-art recommender systems. *Collaborative and Social Information Retrieval and Access-Techniques for Improved User Modeling* 2009:1–22.
10. Lops P, De Gemmis M, Semeraro G. Content-based recommender systems: state of the art and trends. In *Recommender Systems Handbook*. Springer, 2011.
11. Bobadilla J, Ortega F, Hernando A, Gutiérrez A. Recommender systems survey. *Knowledge-Based Systems* 2013; **46**(0):109 –132.
12. Damljanovic D, Stankovic M, Laublet P. Linked data-based concept recommendation: comparison of different methods in open innovation scenario. In *The Semantic Web: Research and Applications*, Simperl E, Cimiano P, Polleres A, Corcho O, Presutti V (eds)., Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012.
13. Felfernig A, Jeran M, Ninaus G, Reinfrank F, Reiterer S. Toward the next generation of recommender systems: applications and research challenges. In *Multimedia Services in Intelligent Environments*. Springer, 2013.
14. Dell'Aglio D, Celino I, Cerizza D. Anatomy of a semantic web-enabled knowledge-based recommender system. *CEUR Workshop Proceedings*, Vol. 667, 2010; 115–130.
15. Cruzes DS, Dybå T. Recommended steps for thematic synthesis in software engineering. *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement*, ESEM '11, IEEE Computer Society, Washington, DC, USA, 2011; 275–284.
16. Mirizzi R, Di Noia T. From exploratory search to web search and back. *Proceedings of the 3rd Workshop on PhD Students in Information and Knowledge Management*, PIKM '10, ACM, New York, NY, USA, 2010; 39–46.
17. Beel J, Langer S, Genzmehr M, Gipp B, Breitinger C, Nürnberger A. Research paper recommender system evaluation: a quantitative literature survey. *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*: ACM, 2013; 15–22.
18. Küster U, König-Ries B. Measures for benchmarking semantic web service matchmaking correctness, The Semantic Web: Research and Applications, 2010; 45–59.
19. Baumann S, Schirru R, Streit B. Towards a storytelling approach for novel artist recommendations. In *Adaptive Multimedia Retrieval. Context, Exploration, and Fusion*, Detyniecki M, Knees P, Nrnberger A, Schedl M, Stober S (eds)., Lecture Notes in Computer Science. Springer Berlin Heidelberg, January 2011.
20. McShane M, Nirenburg S, Beale S. NLP with reasoning and for reasoning. In *Ontology and the Lexicon: A Natural Language Processing Perspective*, Huang C, Calzolari N, Gangemi A, Lenci A, Oltramari A, Prevot L (eds). Cambridge University Press: Cambridge, 2010.
21. Urbani J, Maassen J, Drost N, Seinstra F, Bal H. Scalable RDF data compression with MapReduce. *Concurrency and Computation: Practice and Experience* 2013; **25**(1):24–39.
22. Passant A. Dbrec—music recommendations using DBpedia. In *The Semantic Web ISWC 2010*, Patel-Schneider PF, Pan Y, Hitzler P, Mika P, Zhang L, Pan JZ, Horrocks I, Glimm B (eds)., Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010.
23. Rodríguez Rocha O, Figueroa C, Vagliano I, Moltchanov B. Linked data-driven smart spaces. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, vol. 8638, Balandin S, Andreev S, Koucheryavy Y (eds)., Lecture Notes in Computer Science. Springer International Publishing, 2014 (English).

DOI: 10.1002/cpe

International Journal of Multidisciplinary and Current Research

Research Article

ISSN: 2321-3124
Available at: http://ijmcr.com

An Overview: Metacognition in Education

Mohsen Mahdavi

Department of English, Chabahar Maritime University, Chabahar, Iran

Accepted 01 May 2014, Available online 20 May 2014, **Vol.2** (**May/June 2014 issue**)

Abstract

Metacognition refers to "thinking about thinking" or our ability to know what we know, what we don't know and how to regulate as well as control such thinking. This article seeks to give an overview of some issues related to metacognition, a construct which received a considerable attention on the part of teaching theoreticians and researchers. It starts with a brief introduction of metacognition and then gives an account of its various definitions and components. The differences between cognition and metacognition are also mentioned. It concludes with some ideas and research findings on the teachability of this construct in different fields of study, especially language education.

Keywords: *Metacognition, Metacognitive knowledge, Metacognitive regulation, Self-regulation, Learner autonomy*

1. Introduction

It is by no means easy to talk about metacognition, an apparently unproblematic thirteen-letter term, and its education, both due to the richness and heterogeneity of theoretical and methodological approaches and due to the vague and slippery nature of the metacognition construct. "Hardly does anyone question the reality or the importance of metacognition" (Schraw & Moshman, 1995, p. 351). Tobias et al. (1999 &2009) argued that metacognition very probably is the most dynamically and actively researched cognitive process in areas of current developmental, instructional, and educational psychology. To put it simply, metacognition refers to "thinking about thinking" or our ability to know what we know and what we don't know (Costa & Kallick, 2009; Livingston, 1997). In actuality, offering a definition of metacognition is much more complex than that and is not that simple. There are considerable debates over what exactly this umbrella term is. It has been considered as a fuzzy concept of multifarious definitions by many researchers (Flavell, 1981).

Beyond dispute, the seeds for research programs and development in metacognition were planted and begun to germinate by John Flavell, the pioneer of the field, who deserves great credit for highlighting the depth of his knowledge on metacognition in his landmark pioneering publications on the subject. Metacognition was characterized by Flavell as a "promising new area of investigation" (1979, p. 906). Thereafter, a multitude of empirical and theoretical researches have pursued an agenda on which metacognition was high. Although the term 'metacognition' has not been part of educational psychologists' lexicon and did not come into common use until the 1970s when it was introduced by the aforementioned psychologist. The concept has been around for as long as humans have been able to reflect on their own thinking.

Legitimate grounds exist to heartily endorse a large body of research undertaken on the subject in order to bring unchallenged supremacy of metacognition and give momentum to it as one of the bare essentials to successful learning. To start with, metacognition nurtures independent thinkers and lifelong learners who are able to grapple with new situations and learn how to learn and continue to learn throughout their lifespan in this hectic pace of life (Eggen & Kaucbak, 1995; El-Koumy, 2004; Papaleontiou-Louca, 2003 & 2008; Pilling-Cormick & Garrison, 2007). In the second place, incorporation of metacognition into language teaching can instill a sense of duty and confidence into learners which enables them to self-direct their own learning (Garb, 2000). A necessary step is metacognitive awareness in moving towards learning to regulate learning (Williams & Burden, 1997). The last reason is that metacognition was validated to be central to effective language learning. It is worth emphasizing the point that there is continuing evidence that well-developed metacognitive strategies are the distinguishing quality between good and poor language learners (O'Malley et al., 1989; Gillette, 1990; Rubin, 2005). In the similar vein, Macaro (2001) adds:

Although it is the range and combinations of all strategies that ineffective learners lack, it is the metacognitive … strategies which seem to be the strategy types most lacking in the arsenal of less successful learners." (p. 269) Needless to say, sitting there crosslegged and comfortably waiting hopefully and expecting confidently for learners to automatically "go meta" and self-regulate their own learning seems quite impossible and unrealistic. In a metaphorical sense, "Going meta" connotes becoming an audience for your own performance, that is to say, stepping back to see what you are doing, as though you were someone else actually witnessing it. Learning how to be mindful and manager of one's own learning is not inherited, nor does it happen naturally and overnight, yet it necessitates specific instruction of basic metacognitive skills and strategies. The good news is that metacognitive skills are teachable and learnable as well to build up support for learners to better regulate their cognitive activities (Livingston, 1997; Shannon, 2008; Baer et al., 1994; Brown et al., 1983; Flavell, 1979a; Garner & Alexander, 1989; Borkowski et al., 1987; Bransford et al., 1986; Garner, 1990; Hascher & Oser, 1995). Needed is a big challenge in the howness of instilling and developing metacognition into students in order for helping students learn how to "go meta" concerning mental processes that are not visible directly to create virtuoso performance as learners in their learning experience. Sternberg (2009) contends that:

In the early days, metacognition was more of a curiosity and some psychologists wondered whether it was even a viable construct. Today, I think the question is not whether it is a viable construct, but rather, how it best can be understood, assessed, and developed [taught]. (P. ix)

Metacognition currently carves a unique and successful niche in the self-regulatory phylum and its instruction is a highly flexible and an indispensable approach to language education in that more proficient language learners are more metacognitive than less proficient language learners.

2. Origins and Development

Unquestionably, John Flavell, a developmental psychologist who is now considered to be as the father of the field, was the first one who introduced the term *metacognition* in the 1970s (1971, 1976, 1979). It is defined as "a critical analysis of thought," or simply "thinking about thinking" or "cognition about cognition" (Wellman, 1985; Anderson, 2008; Livingston, 1997). Metacognition can concentrate on any facet of cognition, even metacognition itself (Dunlosky, et al, 2005; Nelson & Narens, 1994). Veenman et al. (2006) regard metacognition as "… a higher-order agent overlooking and governing the cognitive system, while simultaneously being part of it" (p. 5). In his model of cognitive monitoring, Flavell himself offers an early definition of 'metacognition' as:

One's knowledge concerning one's own cognitive processes and products or anything related to them (...) [and] refers, among other things, to the active monitoring and consequent regulation and orchestration of these processes (...), usually in the service of some concrete goal or objective. (Flavell, 1976, p. 232)

What is clear from Flavell's above account, the main constituents of metacognition are "*metacognitive knowledge* and *metacognitive experience* or *regulation*". In addition, he established a link between metacognition and self-regulated learning by making use of the phrase "cognitive monitoring" (Griffith & Ruan, 2005, p. 3). According to Burke (2007), metacognitive skills are sometimes called "self-direction skills" (p. 151).

Based on the proposed model of cognitive monitoring, Flavell held a belief that a wide range of intellectual activities will be monitored by means of the actions and interactions among four basic elements: a) metacognitive knowledge, b) metacognitive experience, c) goals (or tasks), and d) actions (or strategies). *Metacognitive knowledge* refers to one's knowledge or beliefs about person, task, and strategy variables. He has affirmed that metacognitive knowledge is not basically different from other kinds of knowledge in the long-term memory. *Metacognitive experiences* are the segments of this stored knowledge, metacognitive knowledge, that have entered to consciousness, that is, "any conscious cognitive or affective experiences that accompany and pertain to any intellectual enterprise" (Flavell, 1979, p. 906). Metacognitive experiences are very likely to take place in circumstances which requires a great deal of careful, highly 'conscious thinking'. Metacognitive knowledge can be added, deleted, or revised through metacognitive experiences. The *goals or tasks* have to do with the actual objectives of a cognitive endeavor. And finally *actions or strategies*, as the name indicates, are some ways and techniques that may assist in reaching those goals. According to Flavell (1979), acquiring metacognitive strategies as well as cognitive ones is viable. To illustrate the point, Flavell makes some helpful cases of metacognition in real-life experiences

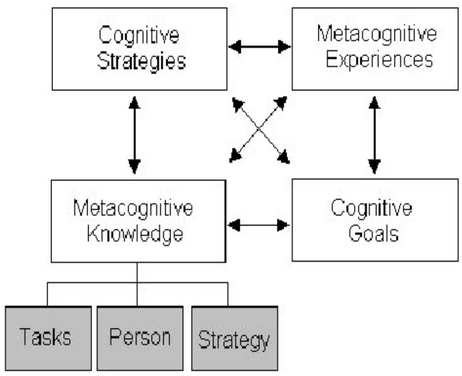


Figure 1: Flavell's model of metacognition (1981, p. 40)

I am engaging in metacognition if I notice that I am having more trouble learning A than B; if it strikes me that I should double-check C before accepting it as a fact; (...) if I become aware that I am not sure what the experimenter really wants me to do; if I sense I had better make a note of D because I may forget it; if I think to ask someone about E to see if I have it right. (Flavell, 1976, p. 232)

Most researchers have now conceptualized metacognition as including two fundamental elements or components referred to as *knowledge of cognition* and *regulation of cognition (*Jacobs & Paris, 1987; Schraw & Moshman, 1995; Schraw, 1998; Brown, 1987; McCormick, 2003; Harris et al., 2010; Williams & Atkins, 2009). Knowledge of cognition refers to knowledge and awareness of one's own cognition. Metacognitive knowledge is "potentially conscious and controllable" (Pressley et al., 1985, p. 4). Moreover, knowledge of cognition or metacognitive knowledge can be stable, usually statable, often fallible, and often late developing information which human as an independent thinker has about his own cognitive process (Baker & Brown, 1984; Garner, 1987; Brown, 1987).

Metacognitive knowledge has been presumably comprised of three distinct, but closely related, facets of knowledge: *declarative*, *procedural*, and *conditional* knowledge (McCormick, 2003; Paris et al., 1983; Harris et al., 2010). Successful coordination and application of these three types of metacognitive knowledge will surely leave its mark on academic development and performance which is heavily contingent upon metacognition (Alexander, 1997; Pressley & Harris, 2006).

Declarative knowledge involves knowledge, skills, and strategies essential for accomplishing a task successfully under various conditions (Hacker, 1998; Pressley & Harris, 2006; Zimmerman & Risemberg, 1997). In other words, it refers to knowing "about things" or "knowing what". Schraw and Moshman (1995) define it as "knowledge about oneself as a learner and about what factors influence one's performance" (p. 352). Flavell (1979) discriminated between kinds of declarative knowledge along the aspects of self or person, task, and strategies or actions.

Procedural knowledge refers to knowledge of how to apply procedures such as learning strategies or actions to make use of declarative knowledge and achieve goals (Harris et al, 2009; Harris et al, 2010; Schraw & Moshman, 1995; Schraw, 1998; McCormick, 2003). It pertains to knowing "how to do things" and "procedures" such as learning strategies. Skilled learners possess more automatic, accurate, and effective procedural knowledge than unskilled learners.

Finally, *conditional knowledge* is referred to as knowledge of when and why to apply various procedures, skills, and cognitive actions or strategies (McCormick, 2003; Schraw & Moshman, 1995; Schraw, 1998; Garner, 1990). Harris et al. (2010) define it as "knowing when, where, and why to use declarative knowledge as well as particular procedures or strategies (procedural knowledge), and is critical to effective use of strategies" (Harris et al., 2009, p.133). In the same way, Garner (1990) held that conditional knowledge is related to knowing when and why to use declarative and procedural knowledge. It is appropriate to add that "[t]he conditional knowledge of successful learners makes them very facile and flexible in their strategy use" (McCormick, 2003, P. 80).

Regulation of cognition or *metacognitive control* is the second major element of metacognition, sometimes also is referred to as executive control, is a sequence of actions taken by students to control their own thinking or learning. It encompasses at least three basic components or essential skills of *planning, monitoring, and evaluation* (Jacobs & Paris, 1987; Schraw & Moshman, 1995; Schraw, 1998).

Planning includes the selection of proper strategies and the provision of resources effective for reaching goals, for instance, making predictions before reading. It includes goal setting, activating prior knowledge, and budgeting time.

Monitoring includes the self-testing skills essential to regulate learning. It refers to the critical analysis of the effectiveness of the strategies or plans being implemented. Schraw (1998) has treated it as "one's on-line awareness of comprehension and task performance" (p.115). Engaging in periodic self-testing in the course of learning would be a particular case of monitoring.

Evaluation refers to the examination of progress being made toward goals which can trigger further planning, monitoring, and evaluation. A typical example might be re-evaluating one's goals and conclusions. To put a fitting end to the discussion on components of metacognition two crucial points are required to be taken into consideration with regard to metacognitive knowledge and metacognitive regulation. Firstly, metacognitive knowledge and experience are related to each other and form partially overlapping sets. Furthermore, they complement and enrich each other. Next, metacognitive knowledge and metacognitive regulation are domain-general in nature and both components appear to embrace a wide spectrum of subject areas and domains.

Gradually, the concept of metacognition underwent some changes and modifications to embrace anything psychological, rather than just anything cognitive (Papaleontiou-Louca, 2003 & 2008). Albeit, when making the first genuine attempt to clearly define the construct of metacognition, Flavell (1979) personally makes reference to the concept as to all those conscious *cognitive* and *affective* experiences that associated with a cognitive enterprise. Flavell (1987) expands the concept of metacognition in a more explicit way to include not only cognitive variables, but rather, anything affective.

In fact, the current literature available on metacognition brings the term to completion by including not only 'thoughts about thoughts', its former definition, but also the following notions: knowledge of one's knowledge, processes, and cognitive and affective states, and the ability to consciously and deliberately monitor and regulate one's knowledge, processes, and cognitive and affective states (Papaleontiou-Louca, 2008).

An important issue which warrants consideration and mention is that the application of knowledge of one's own cognitive and affective processes and the regulation of these processes do not take place in a vacuum, yet, as many theorists and models of metacognition suggest, are highly influenced by one's goals, motivations, perceptions of ability, attributions, and beliefs, as well as context, such as social and cultural norms (Borkowski, et al., 1992; Paris & Winograd, 1990a; Schunk, 1989). Obtaining a full better understanding of metacognition is contingent upon taking these major factors into due consideration as they constitute influences on metacognition as well as being influenced by metacognition (see Borkowski et al., 2000; Pintrich & Zusho, 2002; Zimmerman, 2002).

3. Metacognition versus Cognition

One noteworthy discrimination for fathoming out the true character of the concept of metacognition is to elucidate the distinction between metacognition and cognition (Nelson, 1999; Nelson & Narens, 1994). Nelson (1999) refers to metacognition as "the scientific study of an individual's cognitions about his or her own cognitions" (p. 625). Therefore, metacognition can be considered as a subset of cognition, better to say, a certain kind of cognition. Broadly defined, cognition is a general term for thinking, while metacognition is thinking about thinking.

According to Flavell (1979), metacognition and cognition differ in terms of their content and function, not in their form and quality, i.e., both can be acquired and forgotten, be either correct or incorrect, and so forth. It is safe to say that the aforementioned idea seems an ideal point of departure to draw a sharp distinction between metacognition and cognition. From such a view, the *contents* of metacognition are the knowledge, skills, strategies, and information about cognition, a portion of mental world, while cognition has to do with things in both external and mental world (Amado Gama, 2005). Hacker (1998) articulates that

Metacognitive thoughts do not spring from a person's immediate external reality; rather, their source is tied to the person's own internal mental representations of that reality, which can include what one knows about that internal representation, how it works, and how one feels about it. (Hacker, 1998, p. 3)

From *function* side, cognition acts to resolve problems and bring cognitive activity to a desirable outcome, while metacognitive function is the monitoring and regulation of an individual's cognitive effort in solving a problem and executing a task (Vos, 2001). Cognitive strategies are those strategies which assist a person in accomplishing a particular goal (e.g., comprehending a text), while metacognitive strategies refer to control or regulatory processes such as planning, monitoring, and evaluation, which individuals use to ensure that the particular goal has been met (Livingston, 1997; Rubin, 2005; Garner; 1987). That is to say, "cognitive skills facilitate task achievement, and metacognitive skills help to regulate task achievement" (McCormick, 2003, p. 81).

4. Metacognition, Instruction and Learning

"In teaching me independence of thought, they had given me the greatest gift an adult can give to a child besides love, and they had given me that also." (Courtenacy, 1989, p. 326, cited from Paris & Winograd, 1990a, p. 7)

Although much remains to be learned about metacognition, a topic with an honorable history in psychology and education, without question, the fundamental question "Can metacognition or metacognitive strategies be taught or developed?" which has exercised the minds of researchers for quite a long time is no longer an unanswered question drawing on the strong legacy of the research on the topic, but rather a legitimate question with a satisfactory and definite answer, an *emphatic 'yes'* (Bandura , 1986; Hofer & Yu, 2003; Sperling et al., 2004; Borkowski et al., 1987; Bransford et al., 1986; Garner, 1990; Cromley, 2000; Kuhn et al., 1997; Daley, 2002; Schunk, 1990; Israel, 2007). In instilling metacognitive strategies into students, however, one needs to be cautious and aware that metacognition develops slowly and is difficult to teach (Vos, 2001).

Following the coinage of the term 'metacognition', Flavell (1979) claimed that "increasing the quantity and quality of children's metacognitive knowledge and monitoring skills through systematic training may be feasible as well as desirable" (p. 910). Furthermore, Flavell takes a broad vision regarding metacognitive development and offers a beacon of hope that:

It is at least conceivable that the ideas currently brewing in this area could someday be parlayed into a method of teaching children (and adults) to make wise and thoughtful life decisions as well as to comprehend and learn better in formal educational setting. (Flavell 1979, p. 910)

With regard to the centrality of metacognition to learning, Flavell (1979) contends, though with little empirical evidence, that metacognition plays an important role in varying areas of learning such as oral communication of information, oral persuasion, oral comprehension, reading comprehension, writing, language acquisition, attention, memory, problem solving, social cognition, and various types of self-control and self-instruction (p. 906). According to Sternberg (2009), viability and attainment of metacognition is beyond question, yet the question is how it best can be conceptualized, evaluated, and enhanced. Likewise, Kuhn (2000) asserts that what is perhaps the most significant question which necessitates more investigation is "How can metacognitive development be facilitated?" (p. 180).

The potentiality of increasing meaningfulness of students' learning in various fields has been demonstrated by an enormous body of research (e.g. Biggs, 1986; Hartman, 2001a; Pressley & Ghatala, 1990; Paris & Winograd, 1990b; Brown & Palinscar, 1982). Metacognition "has the potential to empower students to take charge of their own learning and to increase the meaningfulness of students' learning" (Amado Gama, 2005, p. 21), it also encourages learners to 'learn what to do when they don't know what to do' (Wade, 1990; Claxton, 2002). Similarly, Chamot et al. (1999) stated that "metacognition or reflecting on one's own thinking and learning is the hallmark of the successful learner" (p. 2). With regard to metacognitive strategies, with the wisdom of a multitude of research, it is safe to say that the more metacognitive one is, the more strategic and successful one is to be in learning; to be more exact, an individual can pull himself up by his bootstraps in his own lifelong learning (Borkowski et al., 1987; Garner & Alexander, 1989; Pressley & Ghatala, 1990; Schraw & Dennison, 1994). On the value of metacognition, Kuhn (2000) rightly puts that

There would seem few more important accomplishments than people become aware of and reflective about their own thinking and able to monitor and manage the ways in which it is influenced by external sources, in both academic, work, and personal life setting. Metacognitive development is a construct that helps to frame this goal. (p. 181)

Concerning to the instruction and development of metacognition, Papaleontiou-Louca (2003) asserts that "[m]etacognition, like everything else, undoubtedly develops with practice" (p. 17). It is believed that metacognition includes *strategies* for planning, monitoring, and evaluating of language use and language learning which are considered as key elements in developing autonomy (Harris, 2003). If education aimed at helping learners to take charge of their own learning, they have to be able to plan, monitor, and evaluate their learning processes. To do so, they need to be metacognitively aware (Hacker et al., 2009). Ariel (1992) suggests that the aim of metacognitive instruction is to

… develop the sensitivity of students to learning situations, to heighten students' awareness of their own cognitive repertoire and the factors that affect the learning process and contribute to successful learning, to teach strategies for learning, and to develop students' capacity to regulate and monitor their activities. (p. 82).

Just like giving a sick person a useless placebo injection, simply providing learners with answers may enable them to resolve the immediate learning problem. Though, it is not a panacea, just a partial remedy that causes definitely as many problems as it solves. Yet, extolling the virtues of metacognition, many researchers take the view that it has the potential to be seen as a kind of panacea for most learning problems learners may encounter through germination of strategies empowering them to manage their own learning and find out the answers by themselves. "Metacognition can provide students with knowledge and confidence that enables them to manage their own learning and empowers them to be inquisitive and persistent in their pursuits" (Paris & Winograd, 1990a, p. 11).

As pertains to metacognitive development, simply providing learners with highly regimented and structured instruction in metacognitive knowledge without metacognitive experience or quite reverse seems to be insufficient for and does not guarantee the development of metacognitive control and self-regulation (Livingston, 1996 & 1997). Thereby, in fostering a culture of metacognition in learners and classroom settings, the most efficacious approach, though there are several approaches, is the one into which both components of metacognition, namely metacognitive knowledge, and metacognitive regulation are incorporated. One which provides the learners with both knowledge of cognitive processes as well as strategies and together with experience or practice in deploying both cognitive and metacognitive strategies and self-evaluation of the outcomes of their learning.

Anderson (2008) suggested that metacognition in language learning can be divided into five primary and intersecting components: 1. Preparing and planning for learning, 2. Selecting and using strategies, 3. Monitoring learning, 4. Orchestrating strategies, and 5. Evaluating learning. It merits a mention that each of these five components of metacognition is engaged in an interactive process which is not of a linear nature, moving from preparation and planning to evaluation, rather a cyclic one.

McCormick (2003) articulated that "[s]ince it has become clear that metacognitive awareness and skills are a central part of many academic tasks, a critical question for educators is how we foster the development of metacognition in students" (p. 90). Incontrovertibly, a great deal more research is required before one can answer this question with any authority. As a grand finale and conclusion to the discussion in this part, a verbatim quote of Anderson (2008) is worth mentioning.

While learning from a good teacher in a well-structured language program is very important, it is perhaps even more important for these learners to have meaningful learning experiences on their own. Good teachers and well-structured language learning programs cannot possibly teach learners everything they need to know. Getting good results from a study depends on learners' going beyond what teachers and programs provide and developing the kind of metacognitive behavior which will enable them to *regulate their own learning*. (Emphasis added, p. 108)

5. Conclusion

This paper made an attempt to provide a brief overview of metacognition by examining its background and summarizing the relevant literature. It has also outlined some basic features and different components of Metacognition. A summary of research findings on metacognitive strategy training in some areas of education have also been included. Metacognition is a powerful construct in today's educational setting, and its principled teaching can instill a sense of independence and autonomy into learners.

References

[1] Alexander, P. (1997). Mapping the multidimensional nature of domain learning: The interplay of cognitive, motivational, and strategic forces. In M. Maeher, P. Pintrich, Advances in motivation and achievement (Vol. 4), pp. 213–250. Greenwich: CT: JAI Press.

[2] Amado Gama, C. (2005, March). Integrating Metacognition in Interactive Learning Environments, Unpublished Ph.D. Unpublished Thesis, Sussex, the US, University of Sussex.

[4] Anderson, N. J. (2008). Metacognition and good language Learners. In C. Griffith, Lessons from Good Language Learners, pp. 99-109. Cambridge, Cambridge University Press.

[5] Ariel, A. (1992). Education of children and adolescents with learning disabilities. In C. G. Bonds (1992). Metacognition: Developing independence in learning. Clearing House, 66(1), 56-59. New York: Macmillan.

[6] Baer, M., Hollenstein, A., Hofstetter, M., Fuchs, M., and Reber-Wyss, M. (1994). How do expert and novice writers differ in their knowledge of the writing process and its regulation (metacognition) from each other, and what are the differences in metacognitive knowledge between writers of different ages? Annual Meeting of the American Educational Research Association.

[7] Baker, L., Brown, A. L. (1984). Metacognitive skills and reading. In D. P. Pearson, M. Kamil, R. Barr, and P. Monsenthal, Handbook of Reading Research, pp. 353–394. New York, Longman.

[8] Bandura, A. (1986). Social foundations of thought and action: A social cognitive theory. Englewood Cliffs, NJ: Prentice-Hall.

[9] Borkowski, J., Carr, M., and Pressely, M. (1987). "Spontaneous" strategy use: Perspectives from metacognitive theory. Intelligence, 11, 61-75.

[10] Borkowski, J. G. and Groteluschen, A (1992). Expanding the boundaries of cognitive interventions. In B. Y. L. Wong (Eds.), Contemporary intervention research in learning disabilities (pp. 1-21). New York: Springer.

[11] Borkowski, J. G., Chan, L. K., and Muthukrishna, N. (2000). A process-oriented model of metacognition: Links between motivation and executive functioning. In G. Schraw and J. Impara (Eds.), Issues in the measurement of metacognition (p. 42). Lincoln: Buros Institute of Mental Measurements, University of Nebraska.

[12] Bransford, J. D., Sherwood, R., Vye, N. J., and Rieser, J. (1986). Teaching thinking and problem solving. American Psychologist, 41(10), 1078-1089.

[13] Brown, A. L., Palinscar, A. S. (1982). Inducing strategic learning from texts by means of informed, self-control training. Topics in Learning and Learning Disabilities, 5(1): 1–17.

[14] Brown, A. L., Bransford, J. D., Ferrara, R. A., Campione, J. C. (1983). Learning, remembering, and understanding. In JH Flavell, EM Markman, Carmichael's manual of child psychology, pp. 77-166. New York, Wiley.

[15] Brown, A. L. (1987). Metacognition, executive control, self-regulation and other more mysterious mechanisms. In F. E. Weinert and R. H. Kluwe (Eds.), Metacognition, Motivation, and Understanding (pp. 65–116). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

16] Burke, J. (2007). Teaching English Language Arts in a 'Flat' World. Porstmouth, NH: Heinemann.

[17] Chamot, A. U., Brnhardt, S., El-Dinary, P. B., Robin, J. (1999). The learning strategies handbook. White Plains, NY: Addison Wesley Longman.

[18] Claxton, G. (2002). Building learning power. London: TLO Ltd.

[19] Costa, A. L. and Kallick, B. (2009). Leading and Learning with Habits of Mind: 16 Characteristics for Success. Alexandria, VA: Association for Supervision and Curriculum Development.

[20] Cromley, J. (2000). Learning to think: Learning to learn. Washington, DC: National Institute for Literacyo.

[21] Daley, B. J. (2002). Facilitating learning with adult students through concept mapping. Journal of Continuing Higher Education, 50(1), 21-32.

[22] Dunlosky, J., Rawson, K. A., and Middleton, E. L. (2005). What constrains the accuracy of metacomprehension judgments? Testing the transfer-appropriate monitoring and accessibility hypotheses. Journal of Memory and Language, 52, 551-565.

[23] Eggen, P. and Kaucbak D (1995). Strategies for Teachers: Teaching Content and Thinking Skills. Boston: Allyn and Bacon.

[24] El-Koumy, A. S. (2004). Metacognition and Reading Comprehension: Current Trends in Theory and Research. First published by: The Anglo Egyptian Bookshop, 165 Mohamad.

[25] Flavell, J. H. (1976). Metacognitive aspects of problem solving. In L. B. Resnick, The nature of intelligence, pp. 231-23. Hillsdale, NJ: Erlbaum.

[26] Flavell, J. H. (1979a). Metacognition and Cognitive Monitoring: A new area of cognitive developmental inquiry. American Psychologist, 34, 907-11.

[27] Flavell, J. H. (1981). Cognitive monitoring. In W. P. Dickson, Children's Oral Communication (pp. 35–60). New York: Academic Press.

[28] Flavell, J. H. (1987). Speculations about the nature and development of metacognition. In F. E. Weinert, and R. H. Kluwe (Eds.), Metacognition, Motivation and Understanding. Hillsdale, NJ: Earlbaum.

[29] Flavell, J. H. (1971). First discussant's comments: What is memory development the development of?. Human Development, 14(4), 272–278.

[30] Garb, E. (2000). Maximizing the potential of young adults with visual impairments: The metacognitive element. Journal of Visual Impairment and Blindness, 94(9), 574-583.

[31] Garner, R. and Alexander, P. A. (1989). Metacognition: Answered and unanswered questions. Educational Psychologist, 24(2), 143-158.

[32] Garner, R. (1987b). Strategies for reading and studying expository text. Educational Psychologist, 22(3-4), 299-312.

[33] Garner, R. (1990). When children and adults do not use learning strategies: Toward a theory of settings. Educational Research, 60, 517-529.

[34] Gillette, B. K. (1990). Beyond Learning Strategies: A Whole Person Approach to Second Language Learning. Unpublished doctoral dissertation. University of Delaware.

[35] Griffith, P. L. and Ruan, J. (2005). What Is Metacognition and What Should Be Its Role in Literacy Instruction? In S. E. Israel, C. C. Block, K. L. Bauserman, and K. Kinnucan-Welsch, Metacognition in Literacy Learning: Theory, Assessment, Instruction, and Professional Development, pp. 3-18. Mahwah, New Jersey: LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS.

[36] Hacker, D. J. (1998). Self-regulated comprehension during normal reading. In D. J. Hacker, J. Dunlosky, and A. C. Graesser, Metacognition in educational theory and practice, pp. 165-191. Mahwah, NJ: Lawrence Erlbaum Associates.

[37] Hacker, D. J., J. Dunlosky and A. C. Graesser (2009). Handbook of Metacognition in Education. New York: Routledge.

[38] Harris, V. (2003). Adapting classroom-based strategy instruction to a distance learning context. TESL Internet Journal, 7(20).

[39] Harris, K. R., Graham, S., Brindle, M., and Sandmel, K. (2009). Metacognition and Children's Writing. In D. Hacker, J. Dunlosky, and A. C. Graesser, Handbook of Metacognition in Education (pp. 131-153). New York: Routledge.

[40] Harris, K. R., Santangelo, T., and Graham, S. (2010). Metacognition and Strategies instruction in Writing. In H. S. Schneider and W. Waters, Metacognition, Strategy Use, and Instruction, pp. 226-256. London, The Guilford Press.

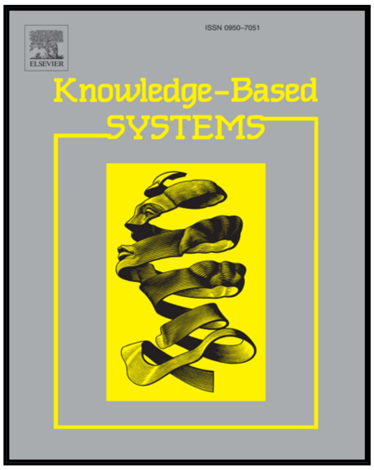
[41] Hartman, H. J. (2001a). Developing students' metacognitive knowledge and strategies. In H. J. Hartman (Eds.), Metacognition in Learning and Instruction: Theory, Research, and Practice,Ch 3 (pp. 33–68). Dordrecht: Kluwer Academic Publishers, The Netherlands.

[42] Hascher ,T. A. and Oser, F. (1995). Promoting autonomy in the workplace: A cognitive-developmental intervention. Paper presented at the Annual Meeting of the American Educational Research Association.

[43] Hofer, B. K. and Yu, S. L. (2003). Teaching self-regulated learning through a "learning to learn" course. Teaching in Psychology, 30(1), 30-33.

[44] Israel, S. E. (2007). Thinking Metacognitively. Using Metacognitive Assessments to Create Individual Reading Instruction. International Reading Association.
[45] Jacob, J. E. and Paris, S. G. (1987). Children's metacognition about reading: Issues in definition, measurement, and instruction. Educational psychologist, 22, 255-278.
[46] Kuhn, D., Shaw, V., and Felton, M. (1997). Effects of dyadic interaction on argumentative reasoning. Cognition and Instruction, 15(3), 287-315.
[49] Kuhn, D. (2000). Metacognitive Development. American Psychological Society, 178-181.
[50] Livingston, J. A. (1996). Effects of metacognitive instruction on strategy use of college students. Unpublished manuscript. State University of New York at Buffalo.
[51] Livingston, J. A. (1997). Metacognition: An Overview. Retrieved from http://www.gse.buffalo.edu/fas/shuell/cep564/Metacog.htm, 1-5.
[52] Macaro, E. (2001). Learning Strategies in Foreign and Second Language Classrooms. London: Continuum.
[53] McCormick, C. B. (2003). Metacognition and Learning. In W. Reynolds, M. Weiner, GE Miller, Handbook of Psychology, pp. 79-102. Hoboken, New Jersey: John Wiley & Sons, Inc.
[54] Nelson, T. O. and Narens, L. (1994). Why investigate metacognition? In J. Metcalfe and A. Shimamura, Metacognition: Knowing about knowing, pp. 1-25. Cambridge, MA: The MIT Press.
[55] Nelson, T. O. (1999). Cognition versus metacognition. In RJ Sternberg, The nature of cognition, pp. 625–641. Cambridge, MA: MIT Press.
[56] O'Malley, J. M., Chamot, A. U., and Kupper, L. (1989). Listening Comprehension Strategies in Second Language Acquisition. Applied Linguistics, 10(2): 418–37.
[57] Papaleontiou-Louca, E. (2003). The Concept and Instruction of Metacognition. Teacher Development, Vol 7, No. 1, 9-30.
[58] Papaleontiou-Louca, E. (2008). Metacognition and Theory of Mind. Cambridge: Cambridge Scholars Publishing.
[59] Paris, S. G., Lipson, M. Y., and Wixson, K. K. (1983). Becoming a strategic reader. Contemporary Educational Psychology, 8, 293–316.
[60] Paris, S. G. and Winograd, P. (1990a). Promoting metacognition and motivation of exceptional children. Remedial and Special Education, 11, 7-15.
[61] Paris, S. G., Winograd P (1990b). How metacognition can promote academic learning and instruction. In B. Idol and L. Jones, Dimensions of thinking and cognitive instruction, pp. 15-51. Hillsdale, NJ: Edrlbaum.
[62] Philling-Cormick, J. and Garrison, D. R. (2007). Self-Directed and Self-Regulated Learning: Conceptual Links. Canadian Journal of University Continuing Education, Vol. 33, No. 2, 13-33.
[63] Pintrich, P. R. and Zusho, A. (2002). The development of academic self-regulation: The role of cognitive and motivational factors. In A. Wigfield and J. S. Eccles (Eds.), Development of achievement motivation (pp. 249-284). San Diego: Academic Press.
[64] Pressley, M., Forrest-Pressley, D. L., Elliott-Faust, D., and Miller, R. (1985). Children's use of cognitive strategies: How to teach strategies and what to do if they can't be taught. In M. Pressley, and J. C. Brained, Cognitive learning and memory in children: Progress in cognitive development research, pp. 1-47. New York, Springer.
[65] Pressley, M. and Ghatala, E. S. (1990). Self-regulated learning: Monitoring learning from text. Educational Psychologist, 25: 19–33.
[66] Pressley, M. and Harris, K. R. (2006). Cognitive strategies instruction: from basic research to classroom instruction. In P. A. Alexander et al. (Eds.), Handbook of educational psychology (pp. 265–286). New York: Macmillan.
[67] Rubin, J. (2005). The Expert Language Learner: a Review of Good Language Learner Studies and Learner Strategies. In K Johnson, Expertise in Second Language Learning and Teaching, pp. 37-63. New York, PALGRAVE MACMILLAN.
[68] Schraw, G. and Dennison, R. S. (1994). Assessing metacognitive awareness. Contemporary Educational Psychology, 19: 460–475.
[69] Schraw, G. and Moshman, D. (1995). Metacognitive Theories. Educational Psychology Review, 7:4, 351-371.
[70] Schraw, G. (1998). Promoting general metacognitive awareness. Instructional Science, 113-125.
[71] Shannon, S. V. (2008). Using Metacognitive Strategies and Learning Styles to Create Self-directed Learners. Institute for Learning Styles Journal, 14-28.
[72] Schunk, D. H. (1989). Social-cognitive theory and self-regulated learning. In D. H. Schunk, B. J. Zimmerman (Eds.), Self-regulated learning and academic achievement: Theory, research and practice (pp. 83-110). New York: Springer.
[73] Schunk, D. H. (1990). Goal setting and self-efficacy during self-regulated learning. Educational Psychologist, 25, 71-86.
[74] Sperling, R. A., Howard, B. C., Staley, R., and Dubois, N. (2004). Metacognition and self-regulated learning constructs. Educational Research and Evaluation, 10(2), 117-139.
[75] Sternberg, R. J. (2009). Foreword to Handbook of Metacognition in Education. In DJ Hacker, JD Dunlosky, AC Graesser, Handbook of Metacognition in Education , pp. viii-ix. Abingdon, Oxon: Routledge.
[76] Tobias, S., Everson, H., and Laitusis, V. (1999, April). Towards a performance based measure of metacognitive knowledge monitoring: Relationships with self-reports and behavior ratings. Paper presented at the annual meeting of the American Educational Research Association. Montreal.
[77] Tobias, S. and Everson, H. T. (2009). The Importance of Knowing What You Know: A Knowledge Monitoring Framework for Studying Metacognition in Education. In D. J. Hacker, J. Dunlosky, and A. C. Graesser, Handbook of Metacognition in Education (pp. 107-127). New York: Routledge.
[78] Veenman, M., Hout-Wolters, V., Bernadette, H. A., and Afflerbach, P. (2006). Metacognition and learning: conceptual and methodological considerations. Metacognition Learning, 1, 3–14.
[79] Vos, H. (2001). Metacognition in Higher Education, PhD thesis. Enschede, The Netherlands, University of Twente.
[80] Wade, S. E. (1990). Using think-aloud to assess comprehension. The Reading Teacher, 43, 7, 442-451.
[81] Wellman, H. (1985). The origins of metacognition. In D. L. Forrest-Pressley, G. E. MacKinnon, and T. G. Waller, Metacognition, Cognition, and Human Performance, volume 1, Theoretical Perspectives, pp. 1–31. Academic Press, Inc.
[82] Williams, J. P. and Atkins J. G. (2009). The Role of Metacognition in Teaching Reading Comprehension to Primary Students. In D. J. Hacker, J. Dunlosky, and A. C. Graesser, Handbook of Metacognition in Education (pp. 26-44). New York: Routledge.
[83] Williams, M. and Burden, R. L. (1997). Psychology for Language Teachers: A social constructivist approach. Cambridge: Cambridge University Press.
[84] Zimmerman, B. J., Risemberg, R. (1997). Becoming a self-regulated writing: A social cognitive perspective. Contemporary Educational Psychology, 22, 73–101.
[85] Zimmerman, B. J. (2002). Becoming a self-regulated learner. Theory into Practice, 41 (2), 65-70.

Accepted Manuscript



Characterizing Context-Aware Recommender Systems: A Systematic Literature Review

Norha M. Villegas, Cristian Sánchez, Javier Díaz-Cely, Gabriel Tamura

PII: S0950-7051(17)30507-5
DOI: 10.1016/j.knosys.2017.11.003
Reference: KNOSYS 4098

To appear in: *Knowledge-Based Systems*

Received date: 14 March 2017
Revised date: 31 October 2017
Accepted date: 2 November 2017

Please cite this article as: Norha M. Villegas, Cristian Sánchez, Javier Díaz-Cely, Gabriel Tamura, Characterizing Context-Aware Recommender Systems: A Systematic Literature Review, *Knowledge-Based Systems* (2017), doi: 10.1016/j.knosys.2017.11.003

Characterizing Context-Aware Recommender Systems: A Systematic Literature Review

Norha M. Villegas[a,*], Cristian Sánchez[a], Javier Díaz-Cely[a], Gabriel Tamura[a]

[a] *Universidad Icesi, Calle 18 No. 122-135, 760031, Cali, Colombia*

Abstract

Context-aware recommender systems leverage the value of recommendations by exploiting context information that affects user preferences and situations, with the goal of recommending items that are really relevant to changing user needs. Despite the importance of context-awareness in the recommender systems realm, researchers and practitioners lack guides that help them understand the state of the art and how to exploit context information to smarten up recommender systems. This paper presents the results of a comprehensive systematic literature review we conducted to survey context-aware recommenders and their mechanisms to exploit context information. The main contribution of this paper is a framework that characterizes context-aware recommendation processes in terms of: i) the recommendation techniques used at every stage of the process, ii) the techniques used to incorporate context, and iii) the stages of the process where context is integrated into the system. This systematic literature review provides a clear understanding about the integration of context into recommender systems, including context types more frequently used in the different application domains and validation mechanisms—explained in terms of the used datasets, properties, metrics, and evaluation protocols. The paper concludes with a set of research opportunities in this field.

Keywords: Recommender systems, Context-aware recommender systems, Pre-filtering, Post-filtering, Context modeling, Recommender systems evaluation

*Corresponding author

Email addresses: `nvillega@icesi.edu.co` (Norha M. Villegas), `cesanchez@icesi.edu.co` (Cristian Sánchez), `jgdiaz@icesi.edu.co` (Javier Díaz-Cely), `gtamura@icesi.edu.co` (Gabriel Tamura)

1. Introduction

With the proliferation of big data & data analytics technologies, recommender systems (RS) are now crucial in seeking customer satisfaction through personalization [1]. RS aim at selecting and proposing the most relevant items, services and offers for their users, by considering their profiles, purchase history, preferences, opinions, interactions with offered products and services, as well as their relationships with other clients. At the same time, the generalization of smart-phones and ubiquitous computing has given RS access to context information [2]. Context-aware recommender systems (CARS) go one step further from traditional RS by exploiting context information such as time, location, and user activity to understand user situations and their influence on user preferences. The incorporation of context information into RS [2, 3] leverages the value of these systems by improving the relevance of possible recommendations with respect to changing user needs [4, 5].

The value of context information to improve the quality of recommendations has been demonstrated and supported by different researchers [6, 7, 8, 9, 10, 11]. Nevertheless, RS as well as context-awareness researchers and practitioners interested in combining the two areas still lack a guide that helps them understand how to exploit context information to smarten up RS. Evidence of this is the absence of comprehensive and domain-independent surveys, particularly systematic literature reviews, that not only consolidate the state of the art of the field, but also explain the most common techniques used to integrate context into the recommendation process. After a rigorous revision of the state of the art, we found that none of the available surveys comprehensively characterize recommendation processes from the perspective of the exploitation of context information. In the best cases, existing surveys focus only on the identification of used context types, and most of them address the problem from the perspective of a particular domain.

This paper presents the findings of a systematic literature review (SLR) [12] on CARS that we conducted with the goal of helping practitioners and researchers understand how context information can be effectively combined with recommendation mechanisms. To this end, we studied a final set of 87 CARS papers that were classified as content-based, collaborative filtering and hybrid approaches. For each paper, we identified recommendation techniques, means to exploit context information, context types, application domains, validation mechanisms including the used datasets, the improvements obtained through the exploitation of context (when measured quantitatively), and research opportunities. The main results of our study are reported in this paper in the form of a framework that characterizes recommendation processes in terms of: i) the recommendation techniques used at every stage of the process, ii) the techniques used to incorporate context, and iii) the stages of the process where context is integrated into the system. This manuscript aims at providing a clear understanding about where context information is usually integrated into the system, what techniques are available to exploit context information depending on the underlying recommendation approach and the phase of the process where context is included, what context types are more frequently exploited in the different application domains, and what validation mechanisms—explained in terms of the used datasets, properties, metrics and evaluation protocols—are generally used to evaluate the proposed approaches. Last, but not least, the paper discusses research opportunities relevant to CARS.

This paper is structured as follows. Sect. 2 explains foundational concepts on recommender systems and context information. Sect. 3 visits related work by analysing the contributions of our SLR with respect to other surveys published on CARS. Sect. 4 explains the methodology we followed to conduct the SLR. Sects. 5–8 constitute the contributions of this manuscript: Sect. 5 presents the findings of our SLR and the characterization framework for CARS; Sect. 6 reports on the validation methods and datasets identified in the studied approaches; Sect. 7 presents quantitative data, reported in the studied papers, on the improvements obtained from the exploitation of context information; and Sect. 8 summarizes and classifies research opportunities. Finally, Sect. 9 concludes the paper.

2. Background

This section briefly presents the fundamentals of RS, and context information as an enabler to improve the quality of recommendations.

2.1. Recommender systems (RS)

Dating back to the mid 1990s, the first recommender systems emerged by following two well differentiated paths. On the one hand, *content-based recommenders* drew from the fields of document retrieval [13, 14] and user profiling [15] to define a common representation space for describing items and users. User profiles result from the aggregation of items that have been favorably or unfavorably qualified in the past. For a given user, items similar to the user's profile are recommended, without taking into account information from other users. On the other hand, *collaborative filtering recommenders* evolved from contributions in human computer interaction [16, 17], where the preferences and choices of similar users are used as the basis for recommendation.

Each of these two types of systems has advantages and disadvantages. Content-based recommenders are easy to explain and understand, prove a good starting point for item navigation, and allow recommendations for new users and/or items (cold start problem). However, they imply the cumbersome task of thoroughly and explicitly describing all items using a common set of features, do not work with implied content, can only handle complementary item recommendation and, being centered on a single user, do not allow the recommendation of serendipitous items. In contrast, collaborative filtering recommenders are based on the common preferences of crowds of users. Thus, these systems cannot only recommend complementary as well as substitute items, but also surprise users by recommending unusual items. Nevertheless, they are not as transparent on their recommendations, need substantially more user data to work well, and do not provide a way to deal with the cold start problem.

Hybrid recommenders, a third type of RS, provide a middle ground between content-based and collaborative filtering systems, by leveraging their strengths and mitigating their drawbacks.

This categorization of RS was proposed by Adomavicius and Tuzhilin in [6]. Other authors have proposed other types of systems [1, 18]. In particular, we consider case-based and knowledge-based systems to be subtypes of the content-based family, community-based systems to be subtypes of the collaborative filtering family, and demographic recommenders to be either content-based or collaborative filtering systems following a pre-filtering stage where data are partitioned in subsets according to user characteristics.

RS use information from items, users, and preferences. The main source of information is the item by user matrix that stores user preferences for individual items. These preferences can be explicitly stated (e.g., in the form of ratings or likes), or implicitly inferred from the interactions of the user with the system (e.g., purchases, accesses or reads). Content-based recommenders consider additional sources of information in the form of feature vectors describing different characteristics of each item (e.g., category, size, age, brand, author).

The characterization of CARS presented in this paper is driven by the stages of the processes followed by content-based and collaborative filtering systems.

2.2. Context information

Abowd et al. define *context* as "*any information useful to characterize the situation of an entity (e.g., a user or an item) that can affect the way users interact with systems*"[2]. The precision of recommendations may result highly affected by context information [7, 8]. For example, a costumer could be more or less interested in a particular restaurant depending on the day of the week. Contextual information can be defined as static or dynamic [3]. When context is static, recommender applications assume that this information is immutable over time. An example of static context is the birthday of a user. On the contrary, dynamic context changes over time thus highly affecting user current needs. Instances of dynamic context are location, time, and user activity [5].

2.2.1. Context categories

Villegas et al. [5] characterize context along five general categories: individual, location, time, activity, and relational. Other characterizations, which can be instantiated from these general categories, have been proposed for domain specific CARS (e.g., the one proposed by Verbert et al. in [19] for CARS in the learning realm). To identify the context types exploited by the CARS studied in this SLR, we based on the classification of context information proposed by Villegas et al., which is summarized as follows:

- **Individual context:** Corresponds to information observed from independent entities (e.g., users or items) that may share common features. This category can be sub-classified into *natural*, *human*, *artificial*, or *groups of entities*. *Natural context* represents characteristics of living and non-living entities that occur naturally, that is, without human intervention (e.g. weather information). *Human context* describes user behavior and preferences (e.g., user payment preferences). *Artificial context* describes entities that result from human actions or technical processes (e.g., hardware and software configurations used in e-commerce platforms). The last subcategory, *groups of entities*, concerns groups of independent subjects that share common features, and that might relate each other (e.g., preferences of users in the user's social network).

- **Location context:** Refers to the place associated with an entity's activity (e.g., the city where a user lives). This category is sub-classified as *physical* (e.g., the coordinates of the user's location, a movie theater's address, or the directions to reach the movie theater from the costumer's current location), and *virtual* (e.g., the IP address of a computer that is located within a network).

- **Time context:** Corresponds to information such as time of the day, current time, day of the week, and season of the year. Time context can be categorized as *definite* and *indefinite*. *Definite* context indicates time frames with specific begin and end points. *Indefinite* context refers to recurrent events that occur while another situation takes place, so it does not have a defined duration (e.g. a user's session in an e-commerce application).

- **Activity context:** Refers to the tasks performed by entities (e.g., shopping, the task a user does at a particular time).

- **Relational context:** Refers to entity relationships that arise from the circumstances in which the entities are involved [20]. Relational context can be defined as *social* (i.e., interpersonal relations such as associations or affiliations), *functional* (i.e. the usage than an entity makes of another).

2.2.2. Integrating Context into Recommender Systems

Traditional recommender systems rely on information about users and items. In contrast, CARS rely also on context information that is relevant for the recommendation. Therefore, recommendation tasks in context-aware recommender systems can be seen as a function of users, items and context information [8]:

$$f : Users \times Items \times Context \rightarrow R \tag{1}$$

There exists three paradigms to integrate context information into recommender systems, depending on the phase of the recommendation process at which context is processed [8]:

- **Contextual pre-filtering:** Context information is used as a filtering mechanism applied to the data, before the application of the recommendation model.

- **Contextual post-filtering:** Context information is initially ignored, and preferences are computed by applying traditional recommender algorithms on the entire data. The resulting set of recommendations is then filtered according to context information that is relevant to the user.
- **Contextual modeling:** Context information is directly integrated into the recommendation model, for example as part of the preference computation process.

This SLR characterizes CARS by considering these three paradigms to incorporate context into the recommendation process, and the techniques used for this integration.

3. Related work

We found 15 RS surveys published in relevant venues and journals between 2004 and 2016. However, only 7 out of these 15 surveys, published between 2012 and 2014, relate to the improvement of RS through the incorporation of context information. Aiming at providing a comprehensive understanding of the state of the art of this field, our SLR not only follows a well defined research methodology, but also characterizes CARS along all application domains, context types, and techniques reported in the studied literature. Most importantly, we documented the recommendation processes followed by content-based and collaborative filtering CARS, to characterize how these systems exploit context information along all phases of the process. The characterization includes recommendation techniques, paradigms for incorporating context, context types, application domains, and a detailed explanation of the mechanisms used to exploit context. We also compiled a catalog of datasets and validation methods used in the studied approaches, as well as a list of open challenges.

Table 1 compares our literature review (last row) with the most relevant CARS surveys we found in the state of the art. This comparison is based on seven criteria that we define as follows: *i) SLR*, the literature review follows a systematic methodology; *ii) not focused on particular domains or techniques*, the survey reviews the state of the art across all identified domains and techniques; *iii) not focused on particular context types*, the survey reports the exploitation of different context types; *iv) identifies context exploitation techniques*, the survey reports the ways how context was exploited in the studied RS; *v) context in the stages of the recommendation process*, the literature review documents how context is exploited along the stages of the recommendation process; *vi) datasets*, the survey lists the datasets used by the studied systems; and *vii) validation techiques*, the review reports the techniques used to evaluate the studied approaches. The plus sign in a cell indicates that the survey is compliant with the corresponding criterion, whereas the absence of the sign indicates that it is not.

Table 1: Related work—Comparing our SLR with other surveys on CARS

Author/Year	**SLR**	**Not focused on particular domains or techniques**	**Not focused on particular context types**	**Identifies context exploitation techniques**	**Context in the stages of the recom-mendation process**	**Datasets**	**Validation techniques**
Verbert et al., 2012 [19]			+	+			+
Kaminskas and Ricci, 2012 [21]			+	+			+
Liu et al., 2013 [22]			+				
Champiri et al, 2014 [23]			+	+			
Campos et al., 2014 [24]		+					+
Inzunza et al., 2016 [25]	+	+	+				
Seifu and Mogalla., 2016 [26]		+	+				
Our literature review	+	+	+	+	+	+	+

According to Table 1, four surveys focus on particular domains: learning processes [19], music services [21], digital libraries [23], and mobile applications [22]. All surveys identify the different types of context exploited in the studied RS, except the one by Campos et al. [24] that focuses on time context only. Furthermore, this survey does not provide insights on the exploitation of context into RS (context exploitation techniques are not identified), but on the evaluation methods used to evaluate the effectiveness of CARS. The surveys conducted by Verbert et al. [19], and Kaminskas and Ricci [21] describe the techniques used to exploit context in the studied systems and the means used to validate them. However, they focus on particular domains. The survey by Liu et al. [22] focuses only on methods to identify the relevant context and the context types exploited in mobile systems. Thus, besides being domain specific, it does not report on techniques used to take advantage of context. As our literature review, the survey conducted by Inzuza et al. [25] follows a systematic approach and does not relate to a particular application domain, technique or context type. However, it does not report on context exploitation techniques. Also similarly to our work, the work conducted by Seifu and Mogalla [26] aims at characterizing the process followed by CARS in the form of what they call *"a framework of CARS."* Nevertheless, their focus is not the way how context is incorporated and exploited, and the explanation of the framework in their six page paper is not as comprehensive as our characterization. Finally, none of the studied surveys report on the used datasets or relate context and its means to exploit it to the concrete phases of the recommendation process.

4. Methodological aspects

We conducted this study by following the guidelines proposed by Kitchenham and Charters in [12]. With our long-term research goal in mind—*to look for innovative and more effective ways of exploiting context information to improve the effectiveness of recommender systems*, we defined the set of research questions that would allow us to understand the state of the art of CARS. These questions are stated as follows:

- RQ1: How is context information exploited along the recommendation process?
- RQ2: What are the existing techniques used to incorporate context information into RS? For each technique, what are the most common application domains?
- RQ3: Is there any correlation between techniques used to incorporate context into RS and any of the traditional recommendation approaches (i.e., content-based, collaborative filtering and hybrid)?
- RQ4: What are the types of context more commonly exploited by RS? What techniques apply in each case?
- RQ5: What evaluation methods have been used to validate the effectiveness of CARS? What are the most common metrics used by these methods?

To answer these research questions and understand the way how context information is integrated into recommender systems, it was important first to characterize the processes that are followed by these systems, in particular by content-based and collaborative filtering approaches. That is, to understand the data that constitute the inputs, and the stages implemented by each type of recommender system to generate recommendations. This process-oriented characterization allowed us not only to report the techniques and context used by the studied RS, but also to map them to specific phases of the recommendation process, with the goal of leveraging the usefulness of this SLR for understanding the state of the art of this field.

We conducted a bibliographic search of conference proceedings and journal papers published in IEEE, ACM, ScienceDirect, EBSCO and Springer. These databases were selected because of the quality of their publications, and their relevance to RS. We used the search string *(("recommendation systems" OR "recommender systems" OR "recommendation" OR "recommendations") AND ("context aware" OR "context-aware" OR "context information" OR "contextual information" OR "location" OR "social" OR "time" OR "activity" OR "task" OR "environmental"))*.

To select the papers to be included in the study we applied four filters: i) *publication date*, we selected papers published between 2004 and 2016; ii) *publication type, number of citations and language*, we excluded workshop and symposium proceedings, papers with less than 10 citations (with some exceptions for papers recently published) and non-English papers; iii) *relevance*, we studied the abstracts to verify the relevance of each paper. After this third filter, we obtained a total of 286 articles, including surveys on RS.

We thoroughly analyzed all these 286 articles and characterized those proposing CARS according to seven criteria: i) *recommendation system approach*, whether it is content-based, collaborative filtering, or hybrid; ii) *recommendation techniques*, the mechanisms used at the different stages of the recommendation process; iii) *paradigm for incorporating context*, whether it is pre-filtering, post-filtering, or contextual modeling; iv) *context types*, the context categories that are exploited in the recommender system (based on the classification proposed by Villegas and Müller [5]); v) *application domain* (if applicable), the specific area targeted by the proposed RS; vi) *evaluation*, the methods and metrics used to validate the effectiveness of the proposed RS; and vii) *data sets* (when reported), the data used to evaluate the proposed approach.

The fourth and last filter consisted in excluding those papers for which we could not identify any of the mandatory criteria presented above. The final set of papers includes 87 manuscripts that propose CARS and 15 surveys, including four highly relevant papers that were published in 2017.

5. Characterization of Context-Aware RS (CARS)

This section summarizes, for each type of recommender system, the findings of our SLR. We consider that the differences between content-based, collaborative filtering, and hybrid recommenders are too profound to analyze them all together, thus we set to do it independently.

To characterize content-based and collaborative filtering CARS, we first represented their recommendation processes using flow diagrams (cf. Figs. 1 and 2) that allow us to distinguish the different phases they comprise, and identify the points where context information is exploited by the surveyed RS, following either the pre-filtering, post-filtering or contextual modeling paradigms.

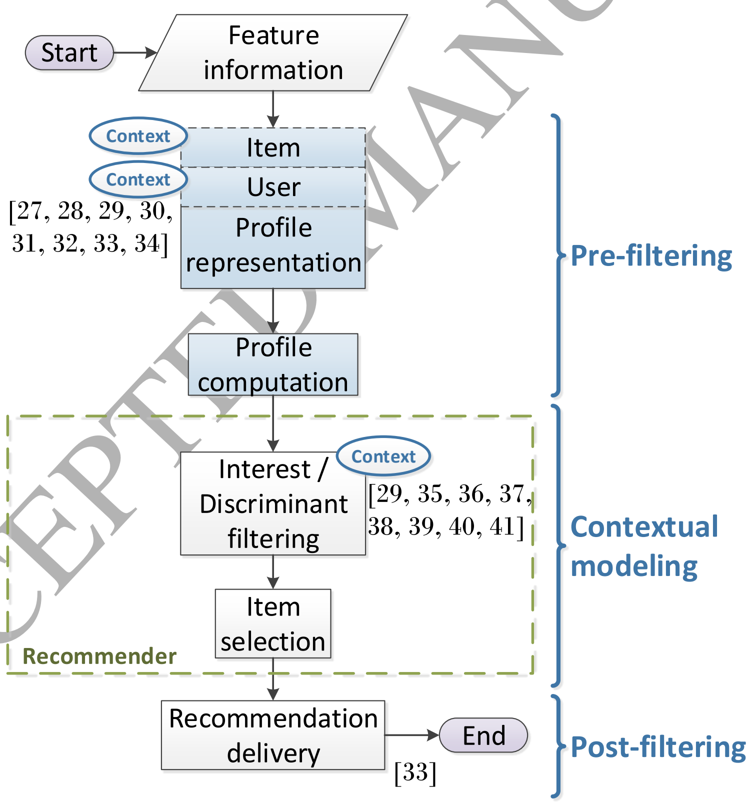
Bold ovals labeled as "Context" indicate the points of the process where we consider that context information can be incorporated. Citations next to each oval correspond to the studied approaches that integrate context in that specific phase of the recommendation process. The absence of citations next to an oval indicates that although we consider context can be exploited at that phase of the process, we found no approaches that do so. Furthermore, each brace depicted in the diagrams groups the stages of the process associated with each of the three paradigms commonly used to incorporate context into a CARS (i.e., pre-filtering, contextual modeling and post-filtering).

It is important to stress out that we focus on the ways in which context can be incorporated and exploited in the recommendation process. Even though on the diagrams we illustrate that process as a whole, we are mainly interested in showing the specific points where the reviewed papers (their references are placed accordingly on the diagrams) decided to adapt the recommendation process to exploit context. While we rely on some of the reviewed papers to illustrate the characterization and findings presented in this section, a more detailed retelling on how each paper implements their system and incorporates context can be found on Tables 2 and 3.

5.1. Content-based approaches

We found 15 papers associated with content-based CARS. Table 2 summarizes the characterization of these papers (cf. Column *Appr.*), which is driven by the process depicted in Fig. 1. Columns *Profile representation*, *Profile computation*, and *Discr. filter* indicate the techniques implemented by the studied systems to realize the main phases of the content-based recommendation process. Column *Paradigm* denotes the strategy used to incorporate context information: pre-filtering, contextual modeling, post-filtering. Column *Context Types* corresponds to the context categories exploited by the corresponding approach. Column *Domains* lists the application domains for which the RS was proposed. The last column explains the means used by the studied CARS to exploit context information.

Figure 1: Process followed by content-based CARS



5.1.1. The beginning of the process

The process implemented by content-based CARS (cf. Fig. 1) begins with the identification of the features in the available data that will define the common dimensional space used to describe item characteristics and user preferences (cf. *User profile definition* and *Item profile definition* in Fig. 1).

Pre-filtering strategies are applicable through the incorporation of contextual factors in the definition of item and/or user profiles. These strategies reduce significantly the search space for the discriminant filter by initially discarding a part of the information available. However, they require the inclusion of redundant user or item profiles for different contextual situations.

All content-based reviewed papers defined the features used as the basis for their recommendation, but only about half of them included contextual information as features. CARS proposed in [27, 28, 29, 30, 31, 32, 33, 34] exploit context using a pre-filtering strategy to generate different contextual *user* profiles for the same user, with different preferences for different situations (see Table 2 for more details regarding the four papers that apply pre-filtering as the paradigm to incorporate context). For instance, [29] proposes a movie CARS where contextual variables of different types such as time (weekday, weekend), location (theater, home), and social context (companion, friends, family) are taken into account to consider or ignore past user ratings, by building several context-aware (micro) profiles that are used to generate context-aware recommendations. As a result, the same user can have different profiles.

None of the surveyed papers associate contextual information with items. We assume that this is because it is easier to think in terms of contextual user profiles than in terms of contextual item profiles, probably because user preferences naturally vary according to context situations. Still, it is completely possible to have different *item* profiles for different situations. Nevertheless, since very often the number of items is many times larger than the number of users, it would mean increasing the complexity of the recommendation process given that a considerably larger number of items must be handled by the system.

5.1.2. The core of the process

The next phase is the core of the recommendation process. In general, a discriminant filter working as a utility function between user and item profiles is responsible for generating a recommendation score from the item and user vectors. This can be done through several strategies: i) by applying some similarity measure such as *Cosine Similarity* (since items and users are represented on the same dimensional feature space, it is possible to compute the distances or similarities between them, with the goal of selecting the items closer to the user's preferences [27, 28, 29, 30, 31, 34, 35, 36]); ii) by obtaining a given classification score by applying a supervised learning technique ([37, 38, 39]); or iii) by applying a heuristic approach (context information can be considered into a discriminant filter, not as additional profile dimensions, but as an integral part of the function definition [32, 33, 35, 40, 41]). Either way, the recommender engine will associate a numeric value to each item, order the items accordingly, and select the ones that appear at the top or that surpass a specified threshold.

At this stage of the process, contextual information can be incorporated by influencing the similarity or distance between items and users. For example, [36] proposes a music recommendation system that incorporates the time at which users accessed different items (songs) in order to provide more relevant recommendations. In their system, users are described by a vector of their correlations to the considered time-related contexts (dawn, morning, monday, tuesday, spring, christmas), items are described as a vector of their correlations to the domain features (e.g., band, genre) as computed by a TF-IDF measure, and the historical accesses to items by users are kept as a collection of pairs of vectors as previously described. To perform a recommendation, the cosine similarity measure is applied to the user's current context and the historical accesses, the similarity of the historical accesses to the items is computed, and an aggregation of both measures allows the scoring of every available songs, so that the top five songs are presented to the user.

5.1.3. The end of the process

Finally, the selected recommendations are organized and delivered to the user. Post-filtering strategies apply at this stage to eliminate the recommendations that are irrelevant to the user's current context. We found that only the RS presented in [33] applied this paradigm to filter out movie recommendations that did not correspond to the current time and location.

5.1.4. Findings

Regarding the paradigm used to incorporate context into the RS (cf. Sect. 2.2.2), findings show that content-based approaches use contextual modeling as much as pre-filtering (one of those combining both strategies);

both paradigms being followed by 53% of the papers. Only one of the studied content-based CARS [33] incorporates context information using post-filtering, combined with pre-filtering. We hypothesize that this may be in part because post-filtering strategies may result in wasting time and computational resources, since the obtained recommendations may become useless after evaluating them with respect to the current context of the user, which is taken into account only at the end of the process. Indeed, pre-filtering approaches provide more benefits in what respects to computational complexity, and contextual-modeling solutions have proven to be more effective for the accuracy of recommendations [4].

With respect to the types of contextual information commonly used in the reviewed systems, and their application domains, we found that time context is commonly used in application domains such as movies and news; location context, in domains associated with movies, music and points of interest; activity context in domains related to movies, music and points of interest; social context in multimedia applications and human context in web services recommendations. It is of particular interest that none of the reviewed content-based CARS target the e-retailing domain, an otherwise popular application domain in traditional RS.

Table 2: Characterization of content-based approaches

Appr.	**Profile representation**	**Profile computation**	**Discr. filter**	**Paradigm**	**Context types**	**Domains**	**Means to incorporate context**
[28]	Item features	Case based reasoning (CBR)	Cosine similarity	Pre-filtering	Time, Location, Activity, Artificial (environment)	Movies, Music, News	Generates a contextual user profile by revising the user's consumption behavior. Then, it uses cosine correlation to measure the similarity between the user contextual profile and the item profile.
[37]	Item features	Heuristic approach	Decision tree algorithm	Cont. Model.	Activity, Human (age, gender)	Indoor Shopping, Points of interest	Proposes a framework where the relationship between user profiles and services under the same context situation are analyzed to infer user preference rules, using the decision tree algorithm.
[27]	Item features structured by a reference ontology	Heuristic approach	Cosine similarity	Pre-filtering	Activity, Time, Location	Movies	Tracks user browsing behavior, and understands user preferences in each particular context. Then, it performs recommendations by means of an aggregation agent that selects the top N items with the highest inferred values.

Table 2: Characterization of content-based approaches

Appr.	Profile representation	Profile computation	Discr. filter	Paradigm	Context types	Domains	Means to incorporate context
[30]	Tag-based features	Heuristic approach	Cosine similarity	Pre-filtering	Time, Location, Activity, Natural (Weather)	Points of interest	Uses a relational Markov network to match the features of Points of Interest (POI) with the current context. POI's features (e.g. outdoor seating, waiter service, dinner) are taken as the inputs to a neural network used to classify the appropriate level of interest (5 categories) of the user for the POI, under the given context situation. The resulting vector that characterizes the POI is then compared to the user vector using cosine similarity.
[29]	Item features	Heuristic approach	Cosine similarity	Pre-filtering, Cont. Model.	Time, Social, Location	Movies	Pre-filtering: Splits user ratings according to the contextual situation in which the preference is expressed, then builds several context-aware (micro) profiles used to infer preferences for new products. Contextual Modeling: Considers context as a weighting factor that influences the recommendation score of a user for a certain item. It combines the non-contextual vector space representation of user preferences with a vector space representation of context, which is built using the pre-filtering approach.
[35]	Latent semantic features	Term frequency inverse document frequency (TF-IDF)	Cosine similarity	Cont. Model.	Location	News	User is defined by the articles read in the past along with his/her location. The system seeks to rank a set of articles that satisfy the geographical location of the user. The preference score is determined by a cosine function ($f(a, l)$) that measures the appropriateness of each article a to a location l.
[38]	Item features	Heuristic approach	Joint probabilistic distribution	Cont. Model.	Activity	Music	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. The first part computes a probability distribution using the Bayes' rule. The second part computes a prior probability that captures the history of user preferences.
[40]	Item features	Heuristic approach	Heuristic approach	Cont. Model.	Location	Indoor shopping	Focuses on mobile recommender systems for assisting indoor shopping by considering location-context. User preferences are calculated through a heuristic approach that integrates three factors: i) time spent in a brand store, ii) frequency of visits to the store, and ii) the matching between the special offers or promotional activities done in the brand store and the user's preferences.
[36]	Item features	Term frequency inverse document frequency (TF-IDF)	Cosine similarity	Cont. Model.	Time	Music	Context refers to the time at which the user listens to a song. The approach predicts user preferences by: i) computing the similarity between the user's current and historical contexts, ii) computing the correlation between historical context and an item, and iii) deriving the expected preference by multiplying measures obtained in i) and ii).

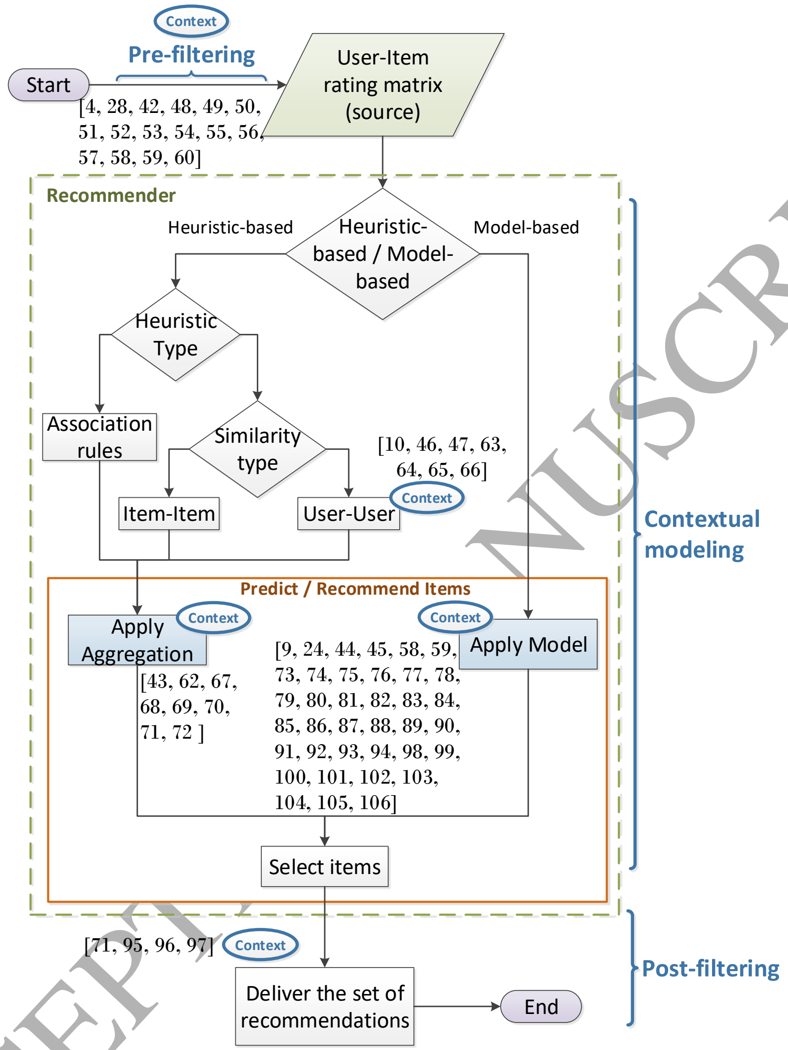
Table 2: Characterization of content-based approaches

Appr.	Profile representation	Profile computation	Discr. filter	Paradigm	Context types	Domains	Means to incorporate context
[39]	Latent semantic features	Heuristic approach	Joint probabilistic distribution	Cont. Model.	Activity, Location	Music	Implements a recommendation model where a set of latent topics is used to associate music content with a user's music preferences under certain location. It is based on the joint probability distribution of user, place, song and lyrics. The latent topics are the intrinsic factors that explain why users prefer certain pieces of music in a particular location and during a specific time period.
[31]	Item features	TF-IDF	Cosine similarity, Jacard similarity	Pre-filtering	Human (user-interest)	Web services	Infers user preferences from the description of the web services that have been accessed by the user.
[32]	Item features	Heuristic	Heuristic	Pre-filtering	Social (followers)	Multimedia	Utilizes Social context (followers) as the basis to decide on user-similarity.
[41]	Item features	Heuristic	Heuristic	Contextual modeling	Location	Points of Interest	Considers context as a weighting factor that influences the recommendation score of a user for a certain item.
[33]	Item property	Heuristic	Heuristic	Pre-Filt, Post-Filt.	Location, Time	Movies	Recommends items with a composite structure (movie theater + movie + showtime). This approach first computes a similarity metric that concerns to the relation between the composite item (theater, movie, showtime) -Pre-Filtering. Then, this similarity measure is incorporated into the discriminant filter -Post-Filtering.
[34]	Item feature	Heuristic	Euclidian Distance	Pre-filtering	Activity	General application	Utilizes a sequential patterns method to find rules from data records on users' smart-phones. Then, by detecting and matching the user's current situation to the rules, which consider his current context and the events in which he has participated, the system determines the most suitable rules for making just-in-time recommendations.

5.2. Collaborative filtering approaches

Figure 2 depicts the general process followed by collaborative filtering CARS. Based on this process, we characterized the 69 collaborative filtering CARS studied in our SLR. This characterization is summarized in Table 3. Column *Recommendation strategy* presents the techniques implemented by the studied approaches, which can follow different paths of the recommendation process, as explained later in this section. As in the characterization of content-based CARS (cf. Table 2), the characterization of collaborative filtering CARS includes the paradigm used to incorporate context into the system (cf. column *Paradigm*), the types of context information exploited by the studied approaches (cf. column *Context Types*), the application domain (cf. column *Domains*) and the mechanisms used to exploit context (cf. column *Means to incorporate context*).

Figure 2: Process followed by collaborative filtering CARS



5.2.1. The beginning of the process

The input of the collaborative filtering process is a user-item rating matrix, where usually rows represent users, and columns represent items. This matrix can include additional dimensions to represent contextual information in the form of synthetic columns or rows, as in the case of the systems presented in [4, 42, 43, 44]. For example, Baltrunas et al. [42] extend the user-item rating matrix into a user-item-context matrix, where contextual information consists in categorical tags (e.g. sunny, cloudy, raining) associated with a given rating.

Depending on the application domain, this matrix can be either obtained directly from the interactions of users with items (e.g., by capturing media accesses instantly [28, 45, 46, 47]), or inferred from historical interactions stored in transactional databases (e.g., by analyzing event logs of previous accesses to the recommended items [10, 48]). This matrix can be very sparse and its processing can be computationally challenging when the number of users and items is considerable (several hundreds of thousands).

At the beginning of the process, *pre-filtering* strategies generate different contextual user-item rating matrices, independent of each other. On the one hand, pre-filtering strategies reduce computational complexity since only a portion of the rating matrix is considered; on the other hand, they imply an extra effort in the acquisition of information, since ratings must be generated for every contextual situation that remains relevant after applying the filter.

We identified 16 papers reporting on the application of context-based pre-filtering strategies to generate recommendations [4, 28, 42, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60]. Pre-filtering is a simple strategy that discards a large part of the data to be analyzed, according to the user's current context. An instance is the process followed by the CARS proposed by Lee et al. [48], in which the authors analyze the access logs to songs, and extract context from the timestamps. Then, they define fuzzy membership functions to fuzzy sets for different contextual variables such as season, time of day, or day of the week, in such a way that the same song recommended at different moments is not considered to be the same item. Another example of collaborative-filtering pre-filtering CARS is the one proposed by Baltrunas et al. [42]: if a statistical test shows that context affects the consumption of an item, they split the item into several synthetic items according to the context situation. For instance, a movie could be split into the same movie associated with winter time, and another one associated with summer time.

5.2.2. The core of the process

To perform the actual recommendation, we identified that most systems apply one of two types of collaborative filtering approaches: *heuristic-based* and *model-based* methods. We found no relationships between any of these methods and particular application domains.

Heuristic-based methods. In the studied systems, heuristic-based approaches are realized through association rules, or the analysis of similarities between users or items. The Apriori algorithm [61] is a common technique for association rule learning. First, it identifies the frequent individual items in the database. Then, it extends them to larger itemsets as long as those appear often enough in the database. Finally, these itemsets are used to determine association rules that allow the discovery of hidden relationships in the data, based on the conditional probability existing between itemsets. The association rules approach is mainly applied to transactional data. However, it can also be applied to the user-item rating matrix, by considering each user row as a single transaction.

An interesting finding of our SLR is that despite approaches such as the one reported in [62] mine association rules, none of the studied systems exploit this technique to incorporate context. A reason for this could be that it would imply extra efforts to acquire the information required to generate a more comprehensive rating matrix, such that the extracted rules are meaningful enough in terms of support, and include context in rule antecedents.

Heuristic-based approaches based on similarity analysis consist in determining the distance between users or items. Each user can be seen as a vector in a feature space with an independent dimension associated with each item (and vice-versa). In general, these distances are determined using neighbourhood or clustering-based methods.

These methods work in two ways. The first one, user-user collaborative filtering, consists in inferring user preferences by determining the group of users that are more similar to the target user, and aggregating the items that are most popular among the members of the user group. The second one, item-item collaborative filtering, consists in determining the similarity among items rated by similar users. In either case, the method requires the computation of the distances between users or items, which can be computationally demanding when dealing with a considerable number of users or items.

Seven of the heuristic-based approaches included in this SLR incorporate context through user-user similarity matching; for instance, the approaches presented in [10, 46, 47, 63, 64, 65, 66] incorporate context to the analysis of user-user similarities (more details can be found on Table 3). On the other hand, none of the heuristic-based approaches use item-item collaborative filtering to incorporate context. As discussed previously, we hypothesize that this is because it results more natural to associate context with users than with items. Nevertheless, in some application domains (e.g., products that are mainly consumed in a particular time of the day), context can be effectively associated with items, in which case an item-item collaborative filtering method that incorporates context would be an appropriate strategy.

Continuing with the recommendation process based on heuristic methods, the information obtained from applying the selected method is aggregated to rank the items to be recommended. Eight of the reviewed papers correspond to collaborative filtering RS that incorporate context as additional factors in the aggregation function. In particular, by using a maximization function [43, 62], a sum of products [67, 68, 69, 70, 71], and probabilities [72]. For instance, Khalid et al. [62] combine the approximated time required to reach a restaurant, the road speed conditions and the distance from the user into a defined metric. Then, the restaurant maximizing this metric is recommended to the user.

Model-based methods. Model-based approaches rely mostly on latent factor models applied to the user-item rating matrix. As we have said before, we can interpret this matrix as either a multi-dimensional representation space where each user is a vector with each item as a dimension, or a multi-dimensional representation space where each item is a vector with each user as a dimension.

The idea of latent factors RS is to obtain a single multi-dimensional space where both users and items can be represented, side by side, through matrix decomposition techniques. In this latent space (usually of smaller dimensionality than the user-item rating space), it is then possible to compute similarities and distances between users and users, users and items, and items and items.

We identified that some systems introduce contextual factors as additional dimensions of the original matrix (e.g., [44, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83]), while some other include contextual information as additive biases on users and items, to affect the calculation of missing ratings (e.g., [9, 45, 84, 85, 86, 87, 88, 89, 90, 91, 92]). An example of the first group is presented in [73], where the authors perform contextual recommendations using tensor factorization. This technique stores the latent feature of users, items and context types in three different matrices. Then, ratings are calculated as the inner product of the latent feature vectors of the given matrices. As a case of the second group, we can consider the RS presented in [85], which performs context-aware recommendations by incorporating temporal changes into the matrix factorization technique. In particular, this approach seeks to capture past temporal patterns over products and items to predict future behaviour, and thus infer preferences. A particular case is the approach presented by Liu et al. [93], which incorporates social context from a social network into the recommendation model by considering that users belonging to different social groups should have different hyperparameters to be used during the matrix factorization process.

It is important to note that despite the collaborative filtering recommendation process indicates that heuristic-based and model-based techniques are not commonly used together, the authors of papers [9] and [88] propose CARS where model-based and heuristic-based techniques are combined. For instance, in [9] user interactions are represented in the form of a social network graph, where each node represents a user, and arc weights correspond to the trust existing between users represented by adjacent nodes (i.e., social context). This approach uses a heuristic-based technique (i.e., graph theory) along with a model-based method (i.e., matrix factorization).

We found a few papers reporting on the application of other approaches. In particular, machine learning techniques, where context information is usually incorporated by implementing probabilistic models such as the Bayesian model [24, 94], or the usage of classifiers such as support vector machines [81, 82, 83].

5.2.3. The end of the process

Similarly to content-based CARS, at the end of the process a contextual filter can be applied to the resulting recommendations to eliminate those items that are irrelevant to the current context. We found four papers reporting on the incorporation of context as a post-filtering strategy to ignore [95], filter [72, 96, 97], or adjust [72, 96] the inferred recommendations.

For example, the systems reported in [72, 96] ignore context until a traditional collaborative filtering algorithm produces restaurant recommendations, which are then adjusted to the user's current context.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[4]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr.:** Sum of products	Pre-filt.	Time, Social, Location	Movies	Filter information according to the current context. A rating is computed for the given user and item, as an aggregation of the ratings of other similar users.
[48]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr.:** Sum of products	Pre-filt.		Music	
[49]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr.:** Top N (most important users)	Pre-filt.		Movies	
[50, 51]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr.:** Sum of products	Pre-filt.	Time, Location	Movies	
[52]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr.:**Sum of products	Pre-filt.	Location	Points of interest	
[28]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr.:**Sum of products	Pre-filt.	Location, Activity, Artificial (environment)	Movies, Music, News	
[53]	**Heuristic-based**, **User & Item sim:** K-medians, **Aggr.:** Maximum	Pre-filt.	Time	E-retailing	
	Heuristic-based, **User & Item sim:** Graph theory, **Aggr.:** Maximum				
[54]	**Heuristic-based**, **User sim:** Graph Theory, **Aggr.:** Maximum	Pre-filt.	Location, Social	Points of Interest	
[60]	**Heuristic-based**, **User & Item sim:** Cosine similarity, **Aggr.:** Sum of products	Pre-filt.	Time	Movies, Music	The authors propose a neighbor-based collaborative filtering approach. A similarity measure over human and time contextual factors provides the basis for estimating the neighborhood of both users and items that will be considered in the recommendation process.
[42]	**Model-based**, **Tech.:** Matrix Fact.	Pre-filt.	Time, Social	Movies	Splits items that have been rated under different context situations. This split is performed only if there is statistical evidence that under these context situations users rate items differently.
[55]	**Model-based**, **Tech.:** Markov Chains	Pre-filt.	Time, Activity	General application	Processes user historical logs to extract contextual features such as day, time range, and location. Then, it identifies common preferences under different contextual conditions. Finally, it makes recommendations based on distributions of user preferences.
[56]	**Heuristic-based**, **User Sim:** Graph theory, **Aggr:** Sum of products	Pre-Filt.	Social	Music, E-retailing	Examines the context-aware recommendation as a search problem in the contextual graph.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[57]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr:** Sum of products	Pre-Filt.	Different types	General Application	Context information associated with users is exploited to infer individual user profiles and from these, the profiles of the groups.
[58]	**Model-based**, **Tech:** Matrix Fact.	Pre-Filt., Cont. Model	Location, Time	Hotels & Tourism	The original user-item rating matrix is divided into sub-matrices according to the temporal states. Then, each sub-matrix is factorized by considering location characteristics.
[59]	**Model based:**, **Tech:** Matrix Fact.	Pre-Filt., Cont. Model.	Location	Web services	Users and services are clustered into groups according to their location. These are then characterized according to their particular QoS features into a local user-service matrix. There is also a global user-service matrix where location is not considered. Matrix factorization is performed on the local and global matrices in a step-wise hierarchical linear process
[46]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr.:** Sum of products	Cont. Model.	Location, Time	Points of interest	Adopts an adjusted Pearson coefficient that computes similarities between users in different contexts. In order to do so, the approach defines a context similarity matrix that includes the coefficient between two users' current contexts for using an item. This coefficient is then incorporated into the aggregation function that computes the missing ratings.
[62]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr.:** Maximum	Cont. Model.	Location, Time	Points of interest	Recommends restaurants by computing the approximate time in reaching it, and considering distance, speed and road conditions. This approximation is included into the aggregation function.
[43]	**Heuristic-based**, **Item sim:** Cosine similarity, **Aggr.:** Maximum **Heuristic-based**, **As. Rules:** Apriori, **Aggr.:** Maximum	Cont. Model.	Location, Time	Points of interest, Music	Transforms the initial user-item matrix by integrating contextual factors as virtual items.
[67]	**Heuristic-based**, **Item sim:** Pearson correlation / Cosine Similarity, **Aggr.:** Sum of products	Cont. Model.	Human (mood), Time	E-learning	
[10]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr.:** Sum of products	Cont. Model.	Time, Human (intent of purchase: Personal-work, Gift Partner, Friend, Parent)	E-retailing	Considers virtual users under different contexts and finds neighbors of contextually similar users to infer recommendations.
[47]	**Heuristic-based**, **User sim:** Jaccard Similarity, **Aggr.:** Sum of products	Cont. Model.	Location, Time	Points of interest	Modifies the Jaccard similarity measure to incorporate context.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[63]	**Heuristic-based**, **User sim:** Pearson Coefficient, **Aggr.:** Sum of products	Cont. Model.	Social	General application	Integrates the strength of the relationships between telecom users into the similarity measure. This strength is modeled taking into account context information associated with phone calls such as duration, time of day and day of the week.
[9]	**Model-based**, **User sim:** Graph theory, **Tech.:** Matrix Factorization	Cont. Model.	Social	Movies	Combines the user-item rating matrix with user-user social contextual information from a trust network to generate a modified rating matrix. This last matrix is then factorized.
[84]	**Model-based**, **Tech.:** Matrix Factorization	Cont. Model.	Social	General application	
[85]	**Model-based**, **Tech.:** Matrix Factorization	Cont. Model.	Time	Movies	Consider context information to add biases on users and items into the recommendation model. Rating values are then influenced by context changes.
[45]			Time	Points of interest	
[86]			Time, Location	Movies	
[87]			Location, Time, Activity	Points of interest	
[91]			Social	Books, Music, Movies	
[92]			Social	General application	
[88]	**Model-based**, **Tech.:** Matrix Factorization	Cont. Model.	Time, Human (Hunger level, mood)	Food, Movies	Clusters items into groups according to the context of their consumption and treats them as virtual items associated with users in a new matrix that is then factorized. Missing ratings are inferred taken into account contextual information.
[89]	**Model-based**, **Tech.:** Matrix Factorization	Cont. Model.	Social	Books, Music, Movies	Considers context information to add biases on users and items into the recommendation model. Through matrix factorization, it creates a common latent factor space for users and items. In this representation space, users and items are clustered independently, so that they can then be brought back to a user-item rating matrix, where missing ratings can be inferred for groups of users.
[90]	**Model-based**, **Tech.:** Matrix Factorization	Cont. Model.	Human (age, gender)	Movies	Constructs several prediction models based on matrix factorization. Each model is then refined by taking into account the predictions from other models. Context information is considered to add biases on users and items into the recommendation model. Rating values are then influenced by context changes.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[73, 74]	**Model-based**, **Tech.:** Tensor Factorization	Cont. Model.	Time, Human, Social	Movies	Perform context-aware recommendations using tensor factorization, which considers the latent features of users and items, and the interaction of the user with an item under a given context. The latent feature of users, items and context types are stored in three matrices. Thus, the inference of preferences is computed as the inner product of the latent feature vectors of the matrices.
[75]			Time	Movies	
[76]			Location, Activity	E-retailing	
[77]			Social, Time	Movies, Food	
[78]			Social, Time	E-retailing, Movies	
[79]			Human (hunger level), Time, Location	Food	
[80]			Social, Time	E-retailing, Movies	
[81, 82]	**Model-based**, **Tech.:** Support Vector Machine (SVD)	Cont. Model.	Time, Social, Natural (weather), Location	Points of interest	Apply SVD to the ratings as represented in a user-item-context space to discriminate between recommended and not recommend items.
[83]	**Model-based**, **Tech.:** Support Vector Machine (SVD)	Cont. Model.	Location	Points of interest	
[94]	**Model-based**, **Tech.:** Bayesian Model	Cont. Model.	Time, Location, Human (mood)	Movies	By adopting a binary particle-swarm optimization technique, identifies the relevant contextual factors for user and item classes, and incorporates them into a latent probabilistic model.
[24]	**Model-based**, **Tech.:** Naïve Bayes	Cont. Model.	Time	Movies	Identifies which members of a household made some specific unidentified ratings of movies by considering time-context conditions such as hour of the day, day of the week and date of rating, as well as number of ratings given by a user. To do this, it analyses temporal trends using probability models.
[98]	**Model-based**, **Tech.:** Sparse Linear Method	Cont. Model.	Time, Location, Social	Movies	Models the contextual rating deviations of items, by assuming that there is a rating deviation for each <item, context condition> pair. This deviation is represented in a matrix, where each row represents an item, and each column represents an individual contextual condition. Then, the ranking score is estimated by an aggregation of user ratings on other items in the same context.
[99]	**Model-based**, **Tech.:** Linear Regression	Cont. Model.	Social, Time	Hotels & Tourism	Predicts user preferences using a linear regression model, which includes a value that represents the user context preference. This value can be computed by means of three different probabilistic methods: i) mutual information based method, ii) information gain based method, and iii) chi-square statistic based method.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[100]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Location, Social	Points of interest, Hotels & Tourism	Location of venues and user social network information are integrated into the matrix factorization model.
[64]	**Heuristic-based**, **Item & User sim:**Pearson correlation, **Aggr:** Weighted ad-hoc	Cont. Model.	Social	Web services	The level of trust among users (social context) is included in the weighted aggregation
[65]	**Heuristic-based**, **User sim:** Ad hoc, **Aggr:** Ad hoc	Cont. Model.	Location, Social	Points of interest, Hotels & Tourism	The social (relationships) and location context of the user is integrated into the process to measure the similarity between users.
[44]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Time, Activity, Location, Artificial	General application	Context-aware preferences as dimensions of the matrix
[93]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model	Social	E-retailing	Social context is considered in order to define groups of users with particular hyper-parameters used by the matrix factorization model
[68]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr:** Sum of Products	Cont. Model	Time, Social	General application	The prediction of user's preference is affected by the user-similiraty , which is computed by considering the context (i.e, the social taggins)
[69]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr:** Sum of Products	Cont. Model	Time	Movies	Adds a time dimension to the original input data. It is defined in a new table which shows item ratings for an active user at different time-frames.
[70]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr:** Sum of products	Cont. Model	Time	Music	Infers user's preference by considering a context score, which is computed for each item in the recommendation list which shows the suitability of that item for the current context of the user.
[101]	**Model-based**, **Tech:** Random walk	Cont. Model.	Social	Social Networks	Tags from social networks are the basis for user similarity (Jaccard). Posts from users are compared by applying an ad-hoc similarity measure. A random walk algorithm is applied in order to estimate weights relating users to users in the social domain and users to items on auxiliary domains (web posts, videos, labels)
[102]	**Model-based**, **Random walk**	Cont. Model.	Time	Web services	Making time-aware personalized QoS prediction is important for high-quality web service recommendation because their performance is highly correlated with invocation time, since service status and network conditions are continuously changing. Time is integrated into a modified Pearson correlation similarity measure (similarities between users and between web services); time is also considered when making the final QoS prediction.
[103]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Social, Time	E-retailing	Social networking features of users (demographics, user posts, groups of related users, temporal activity preferences) that also interact with an unrelated e-commerce site can be transformed into latent factors that can be used for product recommendation, particularly for unknown new users of the e-commerce site.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[104]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Social	Retailing	The authors propose Social Poisson Factorization (SPF) probabilistic model that incorporates social network information into a traditional factorization method, assuming that each user's clicks are driven by their latent preferences for items and the latent influence of their friends (modeled as conditional probabilities). SPF also allows for generating explanations of recommendations based on the social relationships of users.
[105]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Social	Retailing	A probability based matrix factorization is proposed, taking into account trust relationships in a social network in the item recommendation process for retailing purposes. Users and items are then clustered using a Gaussian Mixture Model to enhance the recommendation performance.
[106]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Location, Social	Points of interest	The authors propose a probabilistic matrix factorization method which considers contextual information taken from a location-based social network, where each point of interest is described using a topic model, geographical and social correlations.
[66]	**Heuristic-based**, **User sim:** Jaccard similarity, **Aggr:** Heuristic graph based.	Cont. Model.	Social	Social Networks	The social features of folksonomies are used to provide a user with recommendations of similar users and resources. User profiles consider social contexts, by incorporating information of actions performed by the user on neighboring users' tags, and of other neighboring users on the user's tags. User neighborhoods are defined based on the social network friend relationships according to a specified length of the minimum path linking two users.
[71]	**Heuristic-based**, **User-Sim:**Ad-hoc, **Aggr:** Sum of products	Cont. Model.	Social	E-retailing	Adopts an ad-hoc similarity measure that computes similarities between users in different social context. This measure is then incorporated into the aggregation function that computes the missing ratings
[72]	**Heuristic-based**, **User sim:** Graph theory, **Aggr.:** Probability	Cont. Model., Post-Filt.	Time, Location, Natural (weather), Social	Movies, Hotels & Tourism	Proposes a graph-based contextual model framework. It examines the context-aware recommendation as a search problem in the contextual graph. It also includes a probabilistic-based post-filtering strategy to improve the recommendation results giving contextual factors.
[97]	**Model-based**, **Tech:** Matrix Fact.	Post-Filt.	Time	Movies	The authors propose two successive SVD matrix factorizations to further refine the latent factors for users and items independently, while using time context to filter out unfit items.
[95]	**Heuristic-based**, **User sim:** Cosine Similarity, **Aggr.:** Sum of products	Post-Filt.	Location, Time, Natural (weather)	Hotels & Tourism	Keeps track of contextual features of past user travels to each location. Context aware recommendations are inferred by finding the most similar users, calculating a score for each location, and filtering locations that do not meet contextual conditions.
[96]	**Heuristic-based**, **Users sim:** Pearson correlation, **Aggr.:** Sum of products	Post-Filt.	Time, Location	Points of interest	Adjusts inferred ratings to deliver contextual recommendations.

5.2.4. Findings

The information summarized in Table 3 suggests a correlation between the strategy used to generate recommendations and the paradigm used to incorporate context into the recommendation process of collaborative-filtering CARS.

In general, model-based approaches incorporate context using contextual modeling. This can be explained by the fact that models provide a more natural way to capture interactions between users, items and context. We also found papers reporting on the combination of model-based methods and pre-filtering strategies [42, 55, 58], or even the combination of the three strategies including contextual modelling [59]. However, these combinations may be risky since a pre-filtering strategy can cause loss of valuable information thus affecting accuracy [4].

Heuristic-based approaches are almost evenly distributed between the application of pre-filtering and contextual modeling strategies to realize context-aware recommendations. Regarding the application of pre-filtering, data sources are usually partitioned by context factors to improve data uniformity, which leads to stronger user/item similarities, as well as better confidence and support measures for association rules, thus improving the relevance of recommendations. In the case of contextual modeling, context information modifies how similarity is calculated.

With respect to contextual information, we found that most of the studied collaborative filtering systems have time, social, and location as the predominant factors. Furthermore, the application domains to which the surveyed systems are commonly applied are movies, restaurants, music, points of interests, social networks and e-retailing.

5.3. Hybrid approaches

Since hybrid approaches combine collaborative filtering and content-based recommendation methods in many different ways, there is not a unique abstract process that can characterize hybrid solutions the way we previously did for the non-hybrid processes depicted in Figs. 1 and 2. Table 4 presents the characterization of hybrid approaches, emphasizing on the way context is exploited.

As we found only five papers documenting hybrid RS, it is impossible to generalize their findings. Each approach follows its own strategy.

Table 4: Characterization of hybrid approaches

Appr.	Techniques	Paradigm	Context types	Domains	Means to incorporate context
[107]	**Content-based Profile representation** Item features	Pre-filt.	Time, Location	Movies, Music	Associate ratings with content-based attributes used to describe both user preferences and item features, and with the contextual factors gathered from the user experience (e.g., time of the day). Over the resulting vector space, the authors propose the application of several types of machine learning classification models.
	Collaborative filtering Model-based, **Tech.:** Naïve Bayes, Random forest, Multilayer Perceptron, and Support Vector Machine	Cont. Model.			
[108]	**Collaborative filtering User sim:** K-means	Pre-filt.	Location	Music, Points of interest	Takes into account user demographics: the geographical distance between the user and the event, and the subsequent time that it would take the user to arrive. It segments users into clusters, with every user having a probability of belonging to every cluster, and with each cluster having a probability distribution of liking every item. A discriminant filter evaluates the utility of the item for the user, considering a particular context.
	Content-based Profile representation Item features, **Discr. filter** Heuristic	Cont. Model.			
[28]	**Collaborative filtering User sim:**Pearson correlation, **Heuristic-based** Sum of products	Pre-filt.	Time, Location, Activity, Artificial (environ-ment)	Movies, Music, News	Performs contextual recommendations by combining a discriminant filter with an a aggregation of the ratings of similar users. A similarity measure between users takes into account their contextual profile.
	Content-based Profile representation Item features, **Discr. filter** Cosine similarity	Pre-filt.			
[109]	**Content-based Profile representation:** Item features	Cont. Model.	Social	Web Services	Identifies a couple of reading "experts" whose opinions can be regarded as guidance for news recommendation to particular individuals. Further, integrates this "expert" model with the content information and collaborative filtering, and propose a hybrid recommendation framework.
	Collaborative filtering Model-based, **Tech:** Matrix Factorization	Cont. Model.			
[110]	**Collaborative filtering Model-based**	Cont. Model.	Social, Location, Time	Social Networks	Social context is taken into account by considering the groups to which users belong to on an events-based social network. Users and events are described by the hour at which users attend events (time), and are compared by applying cosine similarity. Geographical preference of events is modeled by obtaining a probability density per user, taking into account the densities of attended events.
	Content-based Profile representation Item features, **Profile comput.** TF-IDF **Discr. filter** Cosine similarity	Pre-filt.			

5.4. Findings in the exploitation of context information

Figure 3 summarizes general findings related to the exploitation of context by the systems described in the surveyed articles.

Figure 3: Summary of findings in the exploitation of context information

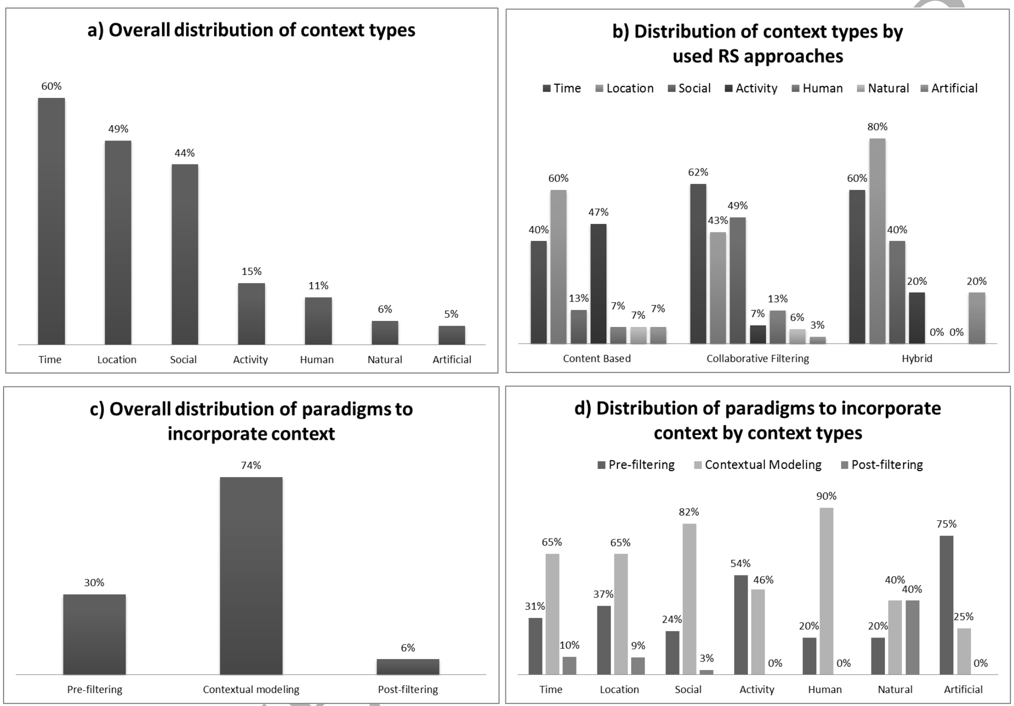


Figure 3.a presents the overall distribution of context types. According to this chart, time is the most used context factor followed by location and social information, whereas artificial is the less exploited context type followed by natural, human and activity. In the studied approaches, artificial context refers to data gathered from mobile sensors, natural context refers to weather conditions, and human context corresponds to user age, gender, mood, intent of purchase, preferences and hunger level. Only papers exploiting social context comment on the reasons why the exploited context type was selected. We hypothesize that, besides being relevant in all application domains, the main reason why time is the most exploited context type is that it is the easiest one to acquire: every system records information about transaction

dates, without requiring the explicit approval of users. As time context, location is also highly relevant and easy to acquire, however, its acquisition and usage, as in the case of social, activity and human context, requires user explicit approval. Artificial context does not necessarily compromise user privacy, however, its acquisition requires physical sensing infrastructures that are not always available.

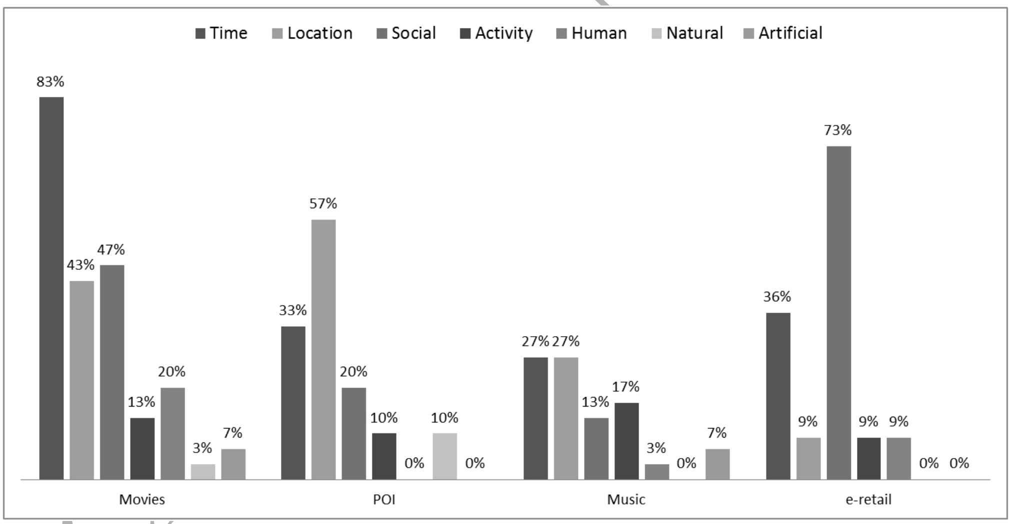
Regarding the context types used with the different recommendation approaches (i.e., content-based, collaborative filtering and hybrid), it is important to highlight that (cf. Fig. 3.b): i) only 13% of the content-based RS exploit social context. This is expected since social context emerges from the relationships among users, which are less relevant in content-based approaches; ii) location and activity are the most used context types in content-based RS. A reason for this is that the relationships existing between users and items usually emerge from the place where the item is used or bought, and the activity the user is performing while using an item. In addition, items are easily associated with places and activities; iii) time is the most exploited context type in collaborative filtering systems. This is probably associated with its easy acquisition, which becomes more relevant in collaborative filtering where it is required to characterize users under similar context situations; and iv) as expected, human context is more relevant in collaborative filtering than in content-based approaches, probably because demographic information is highly used in the analysis of user similarities.

Without doubt, contextual modeling, recognized by its effectiveness in improving the performance of recommendations, is the most common paradigm used to incorporate context into RS (cf. Fig. 3.c). Post-filtering, as discussed in previous subsections, is the less used, since its application may result on the discarding of time and space wise costly recommendations. Concerning the distribution of paradigms to incorporate context by context types (cf. Fig. 3.d), it is worth pointing out that systems exploiting activity (13 papers) and artificial (4 papers) context have pre-filtering as the predominant paradigm to incorporate context.

Most popular application domains identified in the studied papers are movies (30 papers, 34%), points of interest (POI, 18 papers, 21%), music (15 papers, 17%), and e-retailing (11 papers, 13%). Other domains are hotels & tourism (6 papers, 7%), web services (5 papers, 6%), news (4 papers, 5%), food (3 papers, 3%), indoor shopping (2 papers, 2%), social networks (2 papers, 2%), and e-learning (1 paper, 1%). Seven of the studied papers do not report targeting particular application domains (general application).

Figure 4 presents the distribution of context types by application domains. Movies is the only domain that exploits all context types, being time, social, and location the most exploited ones. As expected, location is the most common context type in the points of interest domain, followed by time. Concerning the music domain, location, time and activity are the most used context types. Activity is more predominant in this domain than in the others, probably because music genres are commonly associated with specific user activities. In the e-retailing domain, social is the predominant context type, followed by time. Here it is evident the influence of collaborative filtering as the predominant type of recommendation algorithm, particularly in this domain. Context types location, activity and human are equally exploited in e-retailing applications. Finally, it is worth also noticing that natural context, which in general refers to weather conditions, is more used in points of interest applications.

Figure 4: Distribution of context types by most popular application domains



6. Characterization of validation methods

The improvement of user experience is the ultimate goal of a recommender system. In order to measure it, a series of properties, each with a set of metrics, have been proposed and used since the first developments in the field. These properties allow us to determine the pertinence of the recommendations being suggested. Instances of these properties are predictive power, confidence, diversity, learning rate, coverage, scalability and user evaluation [111].

In this section we summarize the properties that were considered to evaluate the recommendation systems documented in the surveyed papers, particularly predictive power, which is the most commonly used evaluation property. The first two parts of this section focus on prediction metrics and evaluation protocols identified in the studied articles. Then, we summarize other properties that were also used to assess the quality of recommendations in the studied CARS. Finally, we present the list of datasets that we identified in our survey.

6.1. Prediction metrics

Among the different metrics that can be considered to evaluate RS, the most commonly used is predictive power. This could relate to the information retrieval origins of RS. All but five of the papers we surveyed use some kind of prediction metric to assess the quality of their recommendations.

Table 5 presents the distribution of the reviewed articles with respect to prediction metrics. The first column represents the class of metric. The second column refers to the specific prediction metric techniques, grouped by their class. The third column presents the number of papers that use the metric to validate the proposal, which are listed in the last column. It is important to note that some articles may use more than one prediction metric to evaluate their approach. We borrowed the definitions of these metrics from [111] and [112].

Prediction metrics are based on different types of comparisons between the recommended items and the accessed or consumed items. As mentioned in [111], there are three classes of prediction metrics: rating prediction, usage prediction and ranking metrics (cf. first column of Table 5).

Table 5: Metrics used to evaluate predictive power

Class	Prediction Metrics	#Approaches	Approaches' references
Rating prediction metrics	MAE	27	[4, 9, 42, 46, 47, 50, 51, 59, 63, 69, 73, 77, 79, 80, 84, 88, 89, 91, 92, 93, 96, 100, 101, 102, 103, 105, 106]
	RMSE	24	[9, 10, 47, 56, 71, 77, 78, 79, 80, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 100, 101, 105, 106]
Usage prediction metrics	Precision	43	[4, 10, 24, 27, 28, 29, 32, 33, 36, 39, 40, 41, 42, 43, 44, 47, 51, 53, 54, 56, 58, 60, 62, 63, 65, 66, 67, 70, 71, 72, 75, 80, 82, 83, 84, 95, 97, 98, 101, 103, 106, 107, 109]
	Recall	28	[4, 10, 29, 33, 40, 42, 43, 44, 50, 53, 54, 58, 62, 63, 65, 67, 70, 71, 72, 75, 82, 84, 97, 98, 101, 103, 106, 109]
	F-measure	10	[4, 10, 29, 33, 40, 43, 57, 62, 67, 97]
	AUC	5	[24, 60, 74, 103, 107]
	MAP	8	[24, 34, 55, 76, 79, 95, 98, 103]
	BR	1	[95]
Ranking metrics	NDCG or DCG	9	[30, 31, 35, 47, 57, 72, 74, 87, 110]
	Hit Ratio	6	[48, 68, 70, 74, 87, 99]
	MRR or CRR	3	[99, 103, 104]
	Map@K	1	[101]
	Rt10	1	[35]
	None or other type reported	5	[37, 45, 49, 52, 81]

Rating prediction metrics. These metrics measure the correctness of the recommendations in terms of their error. The two metrics we identified in the studied articles are *root mean squared error* (RMSE) and *mean absolute error* (MAE). These metrics measure the distance between predicted and real ratings. So, lower values of RMSE and MAE indicate a higher predictive power. Since RMSE squares the error, it tends to penalize large errors more heavily. The choice between RMSE and MAE is at discretion of the developer. For instance, in the movies domain, while in [85] the RS is evaluated by measuring the quality of suggestions using RMSE, giving more importance to larger differences between the predicted and real ratings, in [73] the evaluation is based on MAE, considering a linear approach to measure the errors.

Usage prediction metrics. These metrics are based on different types of proportions between recommended and consumed items, as determined by the contingency table that compares them. The following are the usage prediction metrics that we identified in the surveyed papers:

- *Precision (or true positive rate)* measures the proportion of recommended items that result relevant to the users, that is, those recommended items that the user actually consumes. The CARS proposed in [39] is evaluated with respect to a context-free approach using this metric. This system exploits user location (i.e., a gym, the library, the office, the transportation system) to suggest appropriate songs. The results show that the proposed approach outperforms its baseline (e.g., a precision of 60% and 50%, respectively, in situations where the location context corresponds to the transportation system).

- *Recall (or sensitivity)* measures the proportion of consumed items that were correctly recommended, that is, the fraction of items relevant to the user that were suggested by the system. Recall and precision are usually considered together as two facets of the quality of the recommendation. An example is presented in [53], where precision and recall are used as the basis to show that the greater the cardinality of a set of recommended items is, the higher the value of recall is.

- *Specificity (or true negative rate)* measures the proportion of not recommended items that are irrelevant to the users. This metric was not directly used in any of the surveyed papers, but it is a basis for the definition of other metrics such as AUC, explained below.

- The *F-measure* family of metrics combines precision and recall, allowing for the comparison of different RS using a single metric. Adomavicius et al. [4] use this metric to compare the effects of taking into account independent context factors (i.e., social, time and location), or combinations of them, when predicting user ratings. The results showed that the segments theater-weekend (i.e., location-time), theater (i.e., location), and theater-friends (i.e., location-social) substantially outperform the standard methods in terms of F-measure. They also applied F-measure to show how their approach outperforms regular non-context RS.

- *AUC (or area under the curve*) is a more robust metric that considers the variations between the true positive rate (recall) and the true negative rate (1 - specificity). The movie CARS published in [74] is evaluated using this metric.

- Other usage prediction metrics are refinements of simpler ones, such as *mean average precision* (MAP) [55], or *benefit ratio* (BR) [95]. The latter is defined as the ratio between the number of users who get an improved prediction and the number of users who get a deteriorated prediction.

Ranking metrics. These metrics assume that the utility of a recommended item is proportional to its position in the ordered list of recommendations produced by the RS. The ranking metrics used to evaluate the CARS included in our survey are the following:

- *Normalized discounted cumulative gain (NDCG) and discounted cumulative gain (DCG)* consider that highly ranked relevant objects give more satisfaction than poorly ranked ones. Biancalana et al. [30] use NDCG to compare their CARS performance with the performance of other approaches. They also study the effect on the quality of recommendations, as measured by NDCG, by taking into account different contextual factors separately. Biancalana et al. [30] and Hong et al. [53] argue that CARS produce better results when the number of items to recommend increases.

- *Hit ratio* measures whether a user's target choice appears in the top-K recommendation list. Generally denoted as Hit@K, where K indicates the number of recommended items. Unger et al. [87] find that the use of latent context models provides a noticeable advantage over non-contextual models for almost every value of K. The advantage is greater with small values of K (i.e., ranging from 1 to 4), which means that the latent context model is highly capable of ranking a suggested recommendation according to the user's current context.

- *Mean reciprocal rank (MRR) and Cumulative reciprocal rank (CRR)* evaluate the ranking position of a user's target choice in the recommendation list. Chen and Chen [99] use CRR to evaluate recommendations that take into account location context.

- *Mean average precision (MAP@K)* considers the precision of the first K recommended ranked items. Every item on the list of ranked items contributes to the MAP@K measure of the recommendation proportionally to its position, if they were indeed accessed/consumed by the user for which the recommendations were made. Jiang et al. [101] use this metric, along with other metrics (MAE, RMSE, Precision, Recall, F1 measure) to evaluate the performance of different configurations of their proposed model.

- *Rt10* averages the ratings of the top 10 recommended items. It is used specially in information retrieval. Son et al. [35] show, using the Rt10 metric, that news article recommendations are more effective when considering their particular geographical location.

Finally, from the five papers that do not report the usage of a particular prediction metric, two of them use other mechanisms to evaluate their models. For instance, to evaluate user satisfaction, Hong et al. [37] measure effectiveness and usability, whereas Baltrunas et al. [45] use a standard usability questionnaire. The approach presented in [81] is compared to a baseline model in terms of accuracy without reporting any metrics. However, these authors published the same model in [82] including a quantitative evaluation.

6.2. Evaluation protocols

This subsection presents the different evaluation protocols applied by the authors of the surveyed papers. These protocols define the way data sets are handled and partitioned into training and test sets to evaluate the quality of the recommendations. We found that in all reported cases context was consistently considered as a data set partitioning criterion, and that the baseline approach is usually a context-free RS, or a CARS that follows a different approach than the one being proposed.

Table 6 presents the distribution of reviewed articles with respect to evaluation protocols. The first column lists the evaluation protocols, the second column shows the number of papers that use the protocol to validate the proposed CARS, and the third column specifies each of the corresponding surveyed papers. Papers [33, 41, 56, 103] did not report on the used evaluation protocol.

Table 6: Evaluation Protocols

Evaluation Protocols	#Approaches	Approaches
Holdout or cross-validation	46	[9, 10, 28, 31, 32, 33, 39, 45, 48, 54, 57, 58, 59, 60, 62, 63, 65, 66, 67, 68, 69, 70, 71, 72, 75, 76, 78, 79, 81, 82, 84, 87, 91, 92, 93, 94, 96, 97, 100, 101, 102, 104, 105, 106, 109, 110]
K-fold cross validation	21	[4, 30, 34, 38, 42, 43, 44, 47, 51, 55, 73, 77, 80, 83, 86, 88, 89, 90, 95, 98, 107]
Hypothesis test	5	[27, 35, 40, 46, 99]
Bootstrapping	2	[4, 29]
Simulation	1	[64]
None reported	4	[33, 41, 56, 103]

Holdout or cross-validation. This is one of the most commonly used evaluation protocols. It consists in splitting the dataset into two sets: training (e.g. 70% of the data) and test (30%). The recommendation model/algorithm is trained using the first set, and evaluated using the second one. The training and test data can be obtained in different ways, depending on the application domain and the way context information affects the recommendations. For example, in [39], Cheng and Shen evaluate their music CARS by splitting the data set according to time and location context, before extracting the training and test sets.

K-fold cross-validation. This is a more sophisticated evaluation protocol that consists in partitioning the dataset into a K equally sized groups of items called folds, to then perform a cross-validation evaluation process. One of the folds is chosen as the test set and the union of the other folds as the training set. This process is repeated K times, each time changing the fold used as test set. This evaluation protocol is used to evaluate the CARS presented in [42]: for each fold, the authors compute the MAE, precision and recall metrics, and average their results to then estimate the quality of their recommendation model. The CARS proposed in [86] and [4] apply independent recommendation processes for each relevant context. The authors evaluate the performance of these systems using K-fold cross-validation. This allows them to compare the predicted ratings for each context, and establish the contexts for which the recommendation is more accurate.

Hypothesis test. This protocol uses statistical inference. It is based on the computation of the statistical significance of the differences between the compared CARS. In particular, it is useful to identify whether there is a significant difference between contextual and non-contextual recommendations. The CARS presented in [99] is evaluated using this protocol, where the hypothesis is that user preferences are influenced by contextual factors, and that the proposed recommendation algorithm is capable of capturing such influences. For example, user restaurant preferences may not be influenced solely by aspects such as food quality, value, and service, but also by contextual factors such as location.

Bootstrapping. This protocol relies on random sampling with replacement. That is, a subset of size N is taken from the original data set and then partitioned into training and test data. This process is repeated multiple times, considering always the whole original data set as the basis for the re-sampling. The estimation of the performances of the RS is finally aggregated from the results of each re-sample. For instance, Musto et al. [29] use a bootstrapping-based protocol proposed in [4]. This protocol consists in identifying different possibly overlapping subsets of the dataset based on context types (e.g., establishing a contextual segment composed of time context observations, or another one composed of location context observations). The authors extract 500 random re-samples from their dataset and split them by assigning 29/30th of the items to the training set and 1/30th to the test set. They use precision, recall and F1 as the metrics to evaluate the performance of their system with respect to the different contextual segments.

Simulation. When there is no dataset available upon which to perform the evaluation of the recommendation model, it is possible to generate an artificial synthetic dataset using simulation techniques, based on certain suppositions (e.g. normal distributions). Eirinaki et al [64] applied this method to generate a social network simulating trust relationships between users (social context), and the matrix relating users to items (in their case, web services).

6.3. Other properties

Predictive metrics measure how close predicted preferences are from user real preferences. However, predictive power is not enough to measure whether the recommendation was satisfactory, useful or effective to the users [112]. A recommendation system may be highly accurate, but only for those items for which a recommendation may result useless (e.g., products that the user buys very frequently).

Table 7 presents the approaches that consider properties other than predictive power to evaluate the proposed CARS. The plus sign in a cell indicates that the corresponding property is used to evaluate the CARS proposed in the paper represented by the row (cf. first column of the table). As in the case of the prediction metrics presented above, we borrowed the definition of these properties from [111] and [112].

Table 7: Other properties

Appr.	Learning rate	Confidence	Diversity	Novelty	Coverage	Scalability	Usability
[76]	+						
[98]	+						
[86]	+						
[90]	+	+	+				
[62]		+				+	
[94]			+		+		
[66]				+			
[53]						+	
[79]	+					+	
[99]	+						+
[30]							+
[27]							+
[38]							+

Learning rate. This property measures how fast an algorithm produces good recommendations. Learning rate is also associated with the parameter that determines how fast or slow a recommendation model will converge towards an optimal solution. We found that all of the CARS evaluated through this property are based on model-based strategies (i.e., matrix and tensor factorization, and linear regression), and exploit context information by implementing the contextual modeling paradigm.

Confidence. This property refers to the trustworthiness of the system predictions, and the extend to which they help users make more effective decisions. The work published in [90] uses this property to evaluate, under specific contexts, the quality of several prediction models based on matrix factorization,

Diversity. This property measures how dissimilar are the recommended items among them. It is defined as the opposite of similarity. Zhang et al.

evaluate the quality of their movie CARS in terms of diversity [90]. They argue that a good recommender system is the one that delivers considerable different recommendations, for example, films belonging to different genres.

Novelty. Based on the assertion that the relevance of a recommended item depends not only on its correctness, but also on its novelty. Nocera et al. [66] define an ad-hoc measure that takes into account whether the recommended items were already known to the user (e.g. accesed in the past).

Coverage. This property measures the proportion of items that the system recommends from the universe of available items. Not all of the available items are subject to be recommended. This is the case of collaborative-filtering RS for items that have not been yet consumed or rated by the users. Sitkrongwong et al. measure accuracy and coverage for different contextual factors [94]. They found that, since not every context applies to all items, it is possible to increase the coverage by ignoring some of the relevant contextual factors. Nevertheless, there is a trade-off between accuracy and coverage that can be mitigated by identifying the set of relevant contextual factors for each user and each item separately, instead of identifying the relevant contextual factors for the entire data set.

Scalability. This property refers to the computational capability of the recommender system to handle a growing amount of data. Khalid et al. address this property by storing and processing data on geographically distributed nodes [62]. Shi et al. measure scalability in terms of time complexity [79]. We did not find any relation between context and scalability.

Usability. This property measures the satisfaction of the user with respect to the ease of use of the RS. In [27], Hawalah and Fasli evaluate usability through a questionnaire that asks users to rate a set of statements, including some to evaluate the contextual nature of the system: i) *the items recommended to me matched my interests*, ii) *the items recommended to me took my personal context requirements into consideration*, and iii) *I was only provided with general recommendations.*

6.4. Data sets

Table 8 characterizes the 16 data sets that we identified as publicly available from 32 out of the 87 characterized papers. For each data set, we indicate the papers that use it, the domain, and the supported context types.

Table 8: Data sets identified in the SLR

Appr.	Domain	Brief description	Context types	URL
[73]	Movies	Information about movies, users and ratings.	Human (age, gender)	https://research.yahoo.com
[9, 50, 57, 60, 75, 78, 80, 90, 97]	Movies	MovieLens: information about ratings, users, and items (movies).	Human (age, gender, occupation), Time (day, month, year, hour, minute, second)	http://grouplens.org/datasets/movielens
[72, 86, 88, 94]	Movies	Data set collected for experiments using an on-line application for rating movies. Users fill in a simple questionnaire created to explicitly acquire the contextual information describing the situation during the consumption. It contains records of users, ratings and movies.	Time (season, day type), Location, Natural (weather), Social	http://students.depaul.edu/yzheng8/DataSets.html
[85]	Movies	Provided by the Netflix Prize. It contains records of ratings, users, and movies.	Time	http://www.netflixprize.com
[24]	Movies	CAMRa 2011s MoviePilot Dataset: contains ratings, users, and items.	Time	http://2011.recsyschallenge.com/dataset
[36, 48]	Music	Information about users, artists, bi-directional user-friend relations, and user-listened artist relations	Social, Time (day, month, year)	http://grouplens.org/datasets/hetrec-2011
[54, 58, 62, 65, 100, 106]	Points of interest, Hotels & Tourism	Data set acquired from FourSquare. It contains information places.	Location, Social	https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[68]	General application	Information about users, tagged papers, and tags.	Time, Social	http://www.citeulike.org/faq/data.adp
[69]	Movies	Provided by the Comaq Systems Research Center. Ratings given by users to movies.	Time	http://www.research.compaq.com/SRC/eachmovie
[65]	Points of interest, Hotels & Tourism	Friendship network with information about locations and user check-ins (user, check-in time, latitude, longitude, location)	Social, Location	http://snap.stanford.edu/data/loc-gowalla.html
[57]	General application	Information of ratings given by users to jokes	Human (user preferences)	http://eigentaste.berkeley.edu/dataset/
[71, 91, 93, 105]	E-retailing	Information about reviews of products done by users	Social	http://www.trustlet.org/epinions.html
[70, 93]	E-retailing, Music	Information about reviews of products done by users	Social, Time	https://labrosa.ee.columbia.edu/millionsong/lastfm
[91, 92, 93, 105]	E-retailing, Books, Music, Movies	Information about user reviews and recommendation services for movies, books, and music	Social	http://socialcomputing.asu.edu/datasets/Douban

7. The effect of incorporating context into RS

When conducting an SRL on CARS, a natural question is the level of improvement of RS performance (e.g., in terms of accuracy) obtained from the inclusion of a particular context type into the recommendation process. Nevertheless, answering this question results impractical, given the wide spectrum of recommendation techniques that can be combined with the different context types, through any of the three existing paradigms to include context information into RS. Furthermore, the performance of these systems vary depending on the used dataset and evaluation metrics, which make the results incomparable. For this reason, questions such as *what is the context type that provides the best results for improving recommendations in a particular context domain?* were not included in the set of research questions that drove the development of this SLR.

Despite the limitations to compare the effectiveness of particular context types, we surveyed the impact of incorporating context information into the reported systems. We found that only 36 out of the 87 studied articles quantitatively evaluate the obtained improvements with respect to baseline approaches (cf. Table 9). This constitutes an opportunity for this research community—formal validations and benchmarks of CARS are of paramount importance to advance this field. The systems reported in these 36 papers were all evaluated with respect to at least one baseline approach in terms of accuracy, through any of the metrics listed in Table 5.

Table 9 presents the improvements reported by these papers. For each approach (cf. Column *Appr.*) the table includes the types of context exploited by the corresponding CARS, the application domain, and the improvement obtained for each of the used metrics. The table groups accuracy metrics according to the three metric categories (i.e., usage prediction, rating prediction and ranking prediction), explained in Sect. 6.1. The goal of this table is to report the surveyed information rather than to provide a basis for comparing the improvements obtained in RS when including the different context types.

Table 9: The effect of incorporating context into the RS that were evaluated quantitatively

Appr.	Types of context	Application domains	Usage Prediction				Rating Prediction		Ranking Prediction
			Precision	Recall	F-Measure	MAP	MAE	RMSE	NDCG, Hit Ratio, MRR
[67]	Human(mood), Time	e-learning	2%	2%	5%				
[50]	Time, Location	Movies		22%			32%		

Table 9: The effect of incorporating context into the RS that were evaluated quantitatively

Appr.	Types of context	Application domains	Usage Prediction				Rating Prediction		Ranking Prediction
			Precision	Recall	F-Measure	MAP	MAE	RMSE	NDCG, Hit Ratio, MRR
[90]	Human (age, gender)	Movies						3%	
[99]	Social, Time	Hotels and Tourism							
[79]	Human (hunger level), Time, Location	Food				15%	9%	9%	
[80]	Social, Time	E-retailing, Movies	6%				17%	14%	
[47]	Location, Time	Point of interest	Between 1,7% and 3,1%				9%	4%	
[51]	Time, Location, Social	Movies	10%						
[72]	Time, Location, Natural (weather), Social	Movies, Hotels and Tourism	Between 80% and 200%; and between 16% and 103%						
[29]	Time, Social, Location	Movies			About 10%				
[98]	Time, Location, Social	Movie	Between 2% and 42%			Between 2% and 6%			
[43]	Time, Location	Music, Point of interest	Between 5% and 33%	Between 5% and 33%	Between 5% and 33%				
[75]	Time	Movies		Between 30% and 35%					
[73]	Human(age, gender), Time, Social	Movies					Between 5% and 30%		
[84]	Social	Not Identified	Between 12% and 22%		About 21%			About 24%	
[76]	Location, Activity	e-retailing	About 53%			About 40%			
[87]	Location, Time, Activity	Point of interest							Hit ratio: About 25%
[68]	Time, Social	General application							Hit ratio: Between 34.56% and 35.91%
[93]	Social	E-commerce					About 10%	About 10%	

Table 9: The effect of incorporating context into the RS that were evaluated quantitatively

Appr.	Types of context	Application domains	Usage Prediction				Rating Prediction		Ranking Prediction
			Precision	Recall	F-Measure	MAP	MAE	RMSE	NDCG, Hit Ratio, MRR
[91]	Social	Books, Music, Movies					Between 9% and 18%	Between 7% and 17%	
[58]	Social	Books, Music, Movies	Avg: 73.27 times better	Avg: 73.27 times better					
[69]	Time	Movies					About 5%		
[92]	Social	General application					Avg: 21%	Avg: 18%	
[31]	Human (user interest)	Web services							NDCG: 40%
[32]	Social	Multimedia	About 25%						
[100]	Social, Location	Points of interest, Hotels & Tourism					Best case: 22%	Best case: 35%	
[65]	Social, Location	Points of interest, Hotels & Tourism	Best case: 15%	Best case: 10%					
[110]	Social, Location, Time	Social networks & Tourism							NDCG: 60%
[101]	Social	General application					Between 10% and 27%		
[59]	Location	Web services					Between 2% and 3%		
[102]	Time	Web services					Between 5% and 20%		
[104]	Social	E-retailing							MRR: between 8% and 25%
[56]	Social	E-retailing	Best case: 78%						
[57]	Different types of context	General application	Best case: 78%						DCG: Between 2.5% and 5%
[106]	Social, Location	Points of interest					Best case: 12.6%	Best case: 14.5%	
[105]	Social	E-retailing					Best case: 16.24%	Best case: 16.09%	

8. Research opportunities

This section provides CARS researchers with a list of research opportunities, most of them borrowed from the studied articles. From each paper, we identified, categorized, and analyzed the challenges that authors defined as worthy of future work. Each subsection corresponds to one of the nine challenge categories that we identified: *dynamic context management, context gathering, context reasoning, contextual modeling, problems inherent in RS, CARS evaluation, users in the loop, self-adaptation* and *privacy and ethical considerations.*

8.1. Dynamic Context Management

Traditional CARS assume that context information is immutable over time, even when user situations continuously change. Evidence of this are deal recommendation systems that keep sending offers to the user for events currently happening in her home city, despite she is in a several day business trip that is scheduled in her agenda, and the user's agenda as well as her current location can be easily monitored by modern applications [7]. This static vision of context information causes that RS deliver recommendations that are irrelevant to users, which has negative effects for businesses.

To deal with this dynamic nature of context, CARS must be equipped with runtime mechanisms to identify relevant context and integrate it into the recommendation process dynamically [3, 5]. This implies also to enable RS to manage the life cycle of context information at runtime, for instance, to identify context variables that become relevant or irrelevant, and treat them accordingly. For example, by adapting the recommendation model according to new context variables that may become relevant while the user interacts with the system.

Dynamic context management research in RS includes investigating mechanisms to i) identify context changes that affect the relevance of recommendations; ii) characterize the life cycle and dynamics of context information; and iii) develop situation-aware and self-adaptation mechanisms to enable CARS with the ability to adjust recommendation models at runtime. Among the studied papers, [3, 28, 30, 36, 47, 69, 85] declare dynamic context and its management challenges as a future research area.

The following two categories of research opportunities, context gathering and context reasoning, are completely related to dynamic context management, since they are concrete phases of the context information life cycle [5].

8.2. Context Gathering

Context gathering refers to the process of acquiring context information from the user's environment. When the relevant context is dynamic (e.g., context that changes over time such as the purchase intent of a user), context acquisition requires automatic mechanisms to detect context sources that become available at runtime, and deploy the sensors required to gather this information. Context gathering challenges include: i) the acquisition of context information from non-explicit and non-traditional context sources (e.g., to identify user intents and motivations); and ii) the development of user interfaces that allow the acquisition of relevant context, without requiring user explicit inputing through traditional interfaces. The authors of the following papers highlight the importance of context gathering research [27, 40, 45, 48, 60, 76, 84, 96].

8.3. Context Reasoning

Context reasoning refers to the inference of implicit context facts from raw context [5]. When context is highly dynamic, context management mechanisms must support the addition of reasoning rules dynamically. Context reasoning challenges in RS include: i) inferring context facts from the combination of different context variables; ii) understanding, particularly at runtime, the relationships between context situations and user preferences; and iii) exploiting context available in user profiles effectively. Authors of papers [4, 30, 39, 45, 52, 58, 82, 86, 107, 108] identify context reasoning as a relevant research topic.

8.4. Contextual Modeling

Pre-filtering, contextual modeling and post-filtering are the three existing paradigms to incorporate context into RS. In contextual modeling, context information is directly integrated into the recommendation model, which, in many cases, has been proved to be more effective than pre- and post-filtering approaches. As a result, an important number of researchers investigate how to exploit context information through contextual modeling [4, 24, 30, 41, 43, 51, 56, 68, 73, 79, 81, 86, 91, 105, 106]. Contextual modeling challenges include the development of new techniques and mechanisms to: i) integrate context into traditional recommendation models; ii) improve rating estimation methods by exploiting context; and iii) identify the context variables that must be integrated into the recommendation model.

8.5. Problems inherent in RS

Context information can be also useful to solve specific problems in RS. Such is the case of the cold-start, self-biased recommendations, and sparsity problems. Concerning the cold-start problem, context provides information that allows the characterization of users, even when they are newcomers to the system [38, 39, 93, 109]. Regarding the self-biased problem, an important challenges is to develop mechanisms to prevent the self-influence of frequently recommended items on future recommendations; the approach presented by Nocera et al. [66] deals with this problem using a novelty metric that considers social context. Concerning the sparsity problem, context-dependent matrices could help decrease sparsity by taking into account different subsets of dimensions under particular context situations [56, 59, 88, 92, 93, 101, 102, 109]. For example, to infer user ratings in a department store, instead of taking into account all of the products the user has bought in the past, one could use only those products directly associated with the user's current purchase intent (e.g., vacation planning, back to school season).

8.6. CARS evaluation

The evaluation of new methods and techniques is crucial to advance the state of the art of CARS, and to confidently apply new developments in real life. Major evaluation challenges identified from the studied papers are [29, 36, 46, 51, 64, 69, 76, 107]: i) the investigation of new properties and metrics; ii) the development of benchmarks that facilitate the understanding of approaches that perform better in particular circumstances; iii) the development and documentation of real life experiments in different application domains; and iv) the acquisition of contextual real data to improve the quality of validations.

8.7. Users in the loop

There is an increasing tendency to conceive users as part of software systems, instead of entities that simply interact with systems. This is commonly known as *the integration of users in the loop*. Users can be integrated in the recommendation process, at one or several of its phases, for example, through feedback that can be used to improve recommendations. Users in the loop are also valuable sources of relevant context. An important challenge is to achieve a seamless integration to avoid affecting the natural behavior of the user. This challenge category was explicitly addressed in [50].

8.8. Self-adaptation

Self-adaptive software systems adjust their structure or behavior at runtime to control the satisfaction of functional and non-functional requirements [113]. To achieve these dynamic capabilities, these systems are instrumented with feedback loops that measure outputs and compare them against reference inputs. If the measure output does not correspond with the desired value specified in the reference input, a controller adjusts the target system to obtain better results [114]. An interesting research direction for the advancement of recommender systems is to instrument them with feedback loop-based mechanisms that allow them to self-improve at runtime. Authors of paper [33] highlight self-adaptation as a promising research direction. In particular, they are interested in implementing a feedback mechanism that adjusts the semantic similarity metric at runtime with the goal of improving performance.

8.9. Privacy and ethical considerations

Privacy and ethics are important aspects to be considered in CARS. Several relevant challenges arise from the need to assure these aspects, which is particularly difficult at runtime. For example, whenever a new context source is identified as relevant, how to validate with the user that this information can be used by the system, that this usage is transparent to the user, and that this information will be used only for the purposes approved by the user. Privacy and ethical aspects are of paramount importance to develop confidence and trust in the use of personalization in CARS [27].

9. Conclusions

This paper presented a comprehensive characterization of context-aware recommendation processes and systems, based on the findings of a systematic literature review (SLR) we conducted to survey CARS that were published between 2004 and 2016. This study was conducted with the goal of helping practitioners and researchers understand how context information can be effectively combined with recommendation mechanisms. The main results provide a clear understanding about where context information is usually integrated into the recommendation process, the techniques available to exploit context information depending on the underlying recommendation approach and the phase of the process where context is included, the context types more frequently exploited in the different application domains, and the most common used evaluation mechanisms, including properties, metrics and protocols.

Despite the comprehensiveness of this study, it is unfeasible to conclude about the effectiveness of using particular context types in specific application domains. This is in part because the effect of including context into RS is difficult to generalize given that the results depend on the nature of the used data sets and recommendation approaches. Furthermore, validation methods must be improved to include quantitative measures that allow a more objective evaluation of the proposed approaches—36 out of the 87 studied papers evaluate their systems quantitatively by comparing, against other approaches used as baselines, the improvements obtained with the integration of context information into the recommender system.

Besides the need for improving validation methods, this survey exposes also several research challenges that deserve further investigation. In particular, those related to the need for i) instrumenting CARS with runtime mechanisms to manage context dynamically along its life cycle; ii) developing new techniques to exploit context directly into the recommendation model; iii) exploiting context to solve inherent RS problems, in particular, the cold-start, self-biased recommendations, and sparsity problems; iv) instrumenting RS with self-adaptation capabilities, and v) solving user-oriented issues such as their better integration in the recommendation loop, as well as the privacy and ethical considerations that arise.

Acknowledgments

This work was funded by Universidad Icesi through its institutional research support program.

References

[1] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, Recommender systems handbook, Vol. 1, Springer, 2011.

[2] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, P. Steggles, Towards a better understanding of context and context-awareness, in: Handheld and ubiquitous computing, Springer, 1999, pp. 304–307.

[3] N. M. Villegas, Context management and self-adaptivity for situation-aware smart software systems, Ph.D. thesis, University of Victoria (2013).

[4] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, Incorporating contextual information in recommender systems using a multidimensional approach, ACM Transactions on Information Systems (TOIS) 23 (1) (2005) 103–145.

[5] N. M. Villegas, H. A. Müller, Managing dynamic context to optimize smart interactions and services, in: The smart internet, Springer, 2010, pp. 289–318.

[6] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 734–749.

[7] S. Ebrahimi, N. M. Villegas, H. A. Müller, A. Thomo, SmarterDeals: a context-aware deal recommendation system based on the SmarterContext engine, in: Proc. 2012 Conf. of the Center for Advanced Studies on Collaborative Research, IBM Corp., 2012, pp. 116–130.

[8] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: Recommender systems handbook, Springer, 2011, pp. 217–253.

[9] H. Ma, T. C. Zhou, M. R. Lyu, I. King, Improving recommender systems by incorporating social contextual information, ACM Transactions on Information Systems (TOIS) 29 (2) (2011) 9.

[10] U. Panniello, M. Gorgoglione, Incorporating context into recommender systems: an empirical comparison of context-based approaches, Electronic Commerce Research 12 (1) (2012) 1–30.

[11] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: A survey, Decision Support Systems 74 (2015) 12 – 32.

[12] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. rep., Keele University (2007).

[13] B. Sheth, P. Maes, Evolving agents for personalized information filtering, in: Proc. 9th Conf. on Artificial Intelligence for Applications, IEEE, 1993, pp. 345–352.

[14] K. Lang, Newsweeder: Learning to filter netnews, in: Proc. 12th Int. Conf. on machine learning, 1995, pp. 331–339.

[15] M. Pazzani, D. Billsus, Learning and revising user profiles: The identification of interesting web sites, Machine learning 27 (3) (1997) 313–331.

[16] W. Hill, L. Stead, M. Rosenstein, G. Furnas, Recommending and evaluating choices in a virtual community of use, in: Proc. SIGCHI Conf. on Human factors in computing systems, ACM Press/Addison-Wesley Publishing Co., 1995, pp. 194–201.

[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in: Proc. 1994 ACM Conf. on Computer supported cooperative work, ACM, 1994, pp. 175–186.

[18] R. Burke, Knowledge-based recommender systems, in: Encyclopedia of library and information systems, Marcel Dekker, 2000, p. 2000.

[19] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, E. Duval, Context-aware recommender systems for learning: a survey and future challenges, IEEE Transactions on Learning Technologies 5 (4) (2012) 318–335.

[20] A. Zimmermann, A. Lorenz, R. Oppermann, An operational definition of context, in: Modeling and using context, Springer, 2007, pp. 558–571.

[21] M. Kaminskas, F. Ricci, Contextual music information retrieval and recommendation: State of the art and challenges, Computer Science Review 6 (2) (2012) 89–119.

[22] Q. Liu, H. Ma, E. Chen, H. Xiong, A survey of context-aware mobile recommendations, International Journal of Information Technology & Decision Making 12 (01) (2013) 139–172.

[23] Z. D. Champiri, S. R. Shahamiri, S. S. B. Salim, A systematic review of scholar context-aware recommender systems, Expert Systems with Applications 42 (3) (2015) 1743–1758.

[24] P. G. Campos, F. Díez, A. Bellogín, Temporal rating habits: a valuable tool for rating discrimination, in: Proc. of the 2nd Challenge on Context-Aware Movie Recommendation, ACM, 2011, pp. 29–35.

[25] S. Inzunza, R. Juárez-Ramírez, A. Ramírez-Noriega, User and context information in context-aware recommender systems: A systematic literature review, in: New Advances in Information Systems and Technologies, Springer, 2016, pp. 649–658.

[26] S. Seifu, S. Mogalla, A comprehensive literature survey of context-aware recommender systems, International Journal of Advanced Research in Computer Science and Software Engineering 3 (4) (2016) 40–46.

[27] A. Hawalah, M. Fasli, Utilizing contextual ontological user profiles for personalized recommendations, Expert Systems with Applications 41 (10) (2014) 4777–4797.

[28] A. M. Otebolaku, M. T. Andrade, Context-aware media recommendations for smart devices, Journal of Ambient Intelligence and Humanized Computing 6 (1) (2015) 13–36.

[29] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, Contextual eVSM: A content-based context-aware recommendation framework based on distributional semantics, in: E-Commerce and Web Technologies, Springer, 2013, pp. 125–136.

[30] C. Biancalana, F. Gasparetti, A. Micarelli, G. Sansonetti, An approach to social recommendation for context-aware mobile services, ACM Transactions on Intelligent Systems and Technology (TIST) 4 (1) (2013) 10.

[31] B. Cao, J. Liu, M. Tang, Z. Zheng, G. Wang, Mashup service recommendation based on user interest and social network, in: 2013 IEEE 20th International Conference on Web Services, 2013, pp. 99–106.

[32] Z. Wang, L. Sun, W. Zhu, S. Yang, H. Li, D. Wu, Joint social and content recommendation for user-generated videos in online social network, IEEE Transactions on Multimedia 15 (3) (2013) 698–709.

[33] L. O. Colombo-Mendoza, R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández, J. J. Samper-Zapater, Recommetz: A context-aware knowledge-based mobile recommender system for movie showtimes, Expert Systems with Applications 42 (3) (2015) 1202–1222.

[34] W. Lee, K. Lee, Making smartphone service recommendations by predicting users' intentions: A context-aware approach, Inf. Sci. 277 (2014) 21–35.

[35] J.-W. Son, A. Kim, S.-B. Park, et al., A location-based news article recommendation with explicit localized semantic analysis, in: Proc. 36th ACM SIGIR Int. Conf. on Research and development in information retrieval, ACM, 2013, pp. 293–302.

[36] D. Shin, J.-w. Lee, J. Yeon, S.-g. Lee, Context-aware recommendation by aggregating user context, in: Proc. 2009 IEEE Conference on Commerce and Enterprise Computing, IEEE, 2009, pp. 423–430.

[37] J. Hong, E.-H. Suh, J. Kim, S. Kim, Context-aware system for proactive personalized service based on context history, Expert Systems with Applications 36 (4) (2009) 7448–7457.

[38] X. Wang, D. Rosenblum, Y. Wang, Context-aware mobile music recommendation for daily activities, in: Proc. 20th ACM Int. Conf. on Multimedia, ACM, 2012, pp. 99–108.

[39] Z. Cheng, J. Shen, Just-for-me: An adaptive personalization system for location-aware social music recommendation, in: Proc. Int. Conf. on Multimedia Retrieval, ACM, 2014, p. 185.

[40] B. Fang, S. Liao, K. Xu, H. Cheng, C. Zhu, H. Chen, A novel mobile recommender system for indoor shopping, Expert Systems with Applications 39 (15) (2012) 11992–12000.

[41] M.-H. Kuo, L.-C. Chen, C.-W. Liang, Building and evaluating a location-based service recommendation system with a preference adjustment mechanism, Expert Systems with Applications 36 (2) (2009) 3543–3554.

[42] L. Baltrunas, F. Ricci, Experimental evaluation of context-dependent collaborative filtering using item splitting, User Modeling and User-Adapted Interaction 24 (1-2) (2014) 7–34.

[43] M. A. Domingues, A. M. Jorge, C. Soares, Dimensions as virtual items: Improving the predictive ability of top n recommender systems, Information Processing & Management 49 (3) (2013) 698–720.

[44] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, J. Tian, Mining mobile user preferences for personalized context-aware recommendation, ACM Trans. Intell. Syst. Technol. 5 (4) (2014) 58:1–58:27.

[45] L. Baltrunas, B. Ludwig, S. Peer, F. Ricci, Context relevance assessment and exploitation in mobile recommender systems, Personal and Ubiquitous Computing 16 (5) (2012) 507–526.

[46] T. H. Dao, S. R. Jeong, H. Ahn, A novel recommendation model of location-based advertising: Context-aware collaborative filtering using ga approach, Expert Systems with Applications 39 (3) (2012) 3731–3739.

[47] L. Hong, L. Zou, C. Zeng, L. Zhang, J. Wang, J. Tian, Context-aware recommendation using role-based trust network, ACM Transactions on Knowledge Discovery from Data (TKDD) 10 (2) (2015) 13.

[48] D. Lee, S. E. Park, M. Kahng, S. Lee, S.-g. Lee, Exploiting contextual information from event logs for personalized recommendation, in: Computer and Information Science 2010, Springer, 2010, pp. 121–139.

[49] Y. S. Lee, X. H. Pham, D. N. Trung, J. J. Jung, H. T. Nguyen, Social context-based movie recommendation: A case study on mymoviehistory, in: Context-Aware Systems and Applications, Springer, 2014, pp. 339–348.

[50] L. He, F. Wu, A time-context-based collaborative filtering algorithm, in: Proc. 2009 IEEE Int. Conf. on Granular Computing, IEEE, 2009, pp. 209–213.

[51] Z. Huang, X. Lu, H. Duan, Context-aware recommendation using rough set model and collaborative filtering, Artificial Intelligence Review 35 (1) (2011) 85–99.

[52] A. Chen, Context-aware collaborative filtering system: Predicting the users preference in the ubiquitous computing environment, in: Location-and Context-Awareness, Springer, 2005, pp. 244–253.

[53] W. Hong, L. Li, T. Li, Product recommendation with temporal dynamics, Expert Systems with Applications 39 (16) (2012) 12398–12406.

[54] H. Bagci, P. Karagoz, Random walk based context-aware activity recommendation for location based social networks, in: Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on, IEEE, 2015, pp. 1–9.

[55] K. Yu, B. Zhang, H. Zhu, H. Cao, J. Tian, Towards personalized context-aware recommendation by mining context logs through topic models, in: Advances in Knowledge Discovery and Data Mining, Springer, 2012, pp. 431–443.

[56] M. Mao, J. Lu, G. Zhang, J. Zhang, Multirelational social recommendations via multigraph ranking, IEEE Transactions on Cybernetics PP (99) (2017) 1–13.

[57] W. Wang, G. Zhang, J. Lu, Member contribution-based group recommender system, Decision Support Systems 87 (2016) 80 – 93.

[58] H. Gao, J. Tang, X. Hu, H. Liu, Exploring temporal effects for location recommendation on location-based social networks, in: Proceedings of the 7th ACM conference on Recommender systems, ACM, 2013, pp. 93–100.

[59] P. He, J. Zhu, Z. Zheng, J. Xu, M. R. Lyu, Location-based hierarchical matrix factorization for web service recommendation, in: 2014 IEEE International Conference on Web Services, ICWS, 2014, 2014, pp. 297–304.

[60] L. Zheng, F. Zhu, S. Huang, J. Xie, Context neighbor recommender: Integrating contexts via neighbors for recommendations, Information Sciences 414 (Supplement C) (2017) 1 – 18.

[61] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, in: Proc. 20th Int. Conf. very large data bases, VLDB, Vol. 1215, 1994, pp. 487–499.

[62] O. Khalid, M. U. S. Khan, S. U. Khan, A. Y. Zomaya, Omnisuggest: A ubiquitous cloud-based context-aware recommendation system for mobile social networks, IEEE Transactions on Services Computing 7 (3) (2014) 401–414.

[63] J. Qi, C. Zhu, Y. Yang, Recommendations based on social relationships in mobile services, Systems Research and Behavioral Science 31 (3) (2014) 424–436.

[64] M. Eirinaki, M. D. Louta, I. Varlamis, A trust-aware system for personalized user recommendations in social networks, IEEE Transactions on Systems, Man, and Cybernetics: Systems 44 (4) (2014) 409–421.

[65] J.-D. Zhang, C.-Y. Chow, Core: Exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations, Information Sciences 293 (2015) 163 – 181.

[66] A. Nocera, D. Ursino, An approach to providing a user of a social folksonomy with recommendations of similar users and potentially interesting resources, Knowledge-Based Systems 24 (8) (2011) 1277–1296.

[67] P. Dwivedi, K. K. Bharadwaj, A fuzzy approach to multidimensional context aware e-learning recommender system, in: Mining Intelligence and Knowledge Exploration, Springer, 2013, pp. 600–610.

[68] N. Zheng, Q. Li, A recommender system based on tag and time information for social tagging systems, Expert Systems with Applications 38 (4) (2011) 4575–4587.

[69] S.-H. Min, I. Han, Detection of the customer time-variant pattern for improving recommender systems, Expert Systems with Applications 28 (2) (2005) 189–199.

[70] N. Hariri, B. Mobasher, R. Burke, Context-aware music recommendation based on latenttopic sequential patterns, in: Proceedings of the sixth ACM conference on Recommender systems, ACM, 2012, pp. 131–138.

[71] P. Symeonidis, E. Tiakas, Y. Manolopoulos, Product recommendation and rating prediction based on multi-modal social networks, in: Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, 2011, pp. 61–68.

ACCEPTED MANUSCRIPT

[72] H. Wu, K. Yue, X. Liu, Y. Pei, B. Li, Context-aware recommendation via graph-based contextual modeling and post-filtering, International Journal of Distributed Sensor Networks 2015 (2015) 16.

[73] A. Karatzoglou, X. Amatriain, L. Baltrunas, N. Oliver, Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering, in: Proc. 4th ACM Conf. on Recommender systems, ACM, 2010, pp. 79–86.

[74] A. Rettinger, H. Wermser, Y. Huang, V. Tresp, Context-aware tensor decomposition for relation prediction in social networks, Social Network Analysis and Mining 2 (4) (2012) 373–385.

[75] B. Hidasi, D. Tikk, Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2012, pp. 67–82.

[76] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, N. Oliver, Tfmap: optimizing map for top-n context-aware recommendation, in: Proc. 35th ACM SIGIR Int. Conf. on Research and development in information retrieval, ACM, 2012, pp. 155–164.

[77] S. Rendle, Z. Gantner, C. Freudenthaler, L. Schmidt-Thieme, Fast context-aware recommendations with factorization machines, in: Proc. 34th ACM SIGIR Int. Conf. on Research and development in Information Retrieval, ACM, 2011, pp. 635–644.

[78] B. Zou, C. Li, L. Tan, H. Chen, GPUTENSOR: efficient tensor factorization for context-aware recommendations, Information Sciences 299 (2015) 159–177.

[79] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, Cars2: Learning context-aware representations for context-aware recommendations, in: Proc. 23rd ACM Int. Conf. on Information and Knowledge Management, ACM, 2014, pp. 291–300.

[80] C. Zheng, E. Haihong, M. Song, J. Song, Cmptf: Contextual modeling probabilistic tensor factorization for recommender systems, Neurocomputing.

[81] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, Context-aware SVM for context-dependent information recommendation, in: Proc. 7th MDM Int. Conf. on Mobile Data Management, IEEE, 2006, pp. 109–109.

[82] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, H. Kato, Context-aware ranking method for information recommendation, in: Advances in Communication Systems and Electrical Engineering, Springer, 2008, pp. 319–337.

[83] Y. Omori, Y. Nonaka, M. Hasegawa, Design and implementation of a context-aware guide application for mobile users based on machine learning, in: Knowledge-Based and Intelligent Information and Engineering Systems, Springer, 2010, pp. 271–279.

[84] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, S. Yang, Social contextual recommendation, in: Proc. of the 21st ACM Int. Conf. on Information and knowledge management, ACM, 2012, pp. 45–54.

[85] Y. Koren, Collaborative filtering with temporal dynamics, Communications of the ACM 53 (4) (2010) 89–97.

[86] A. Odić, M. Tkalčič, J. F. Tasič, A. Košir, Predicting and detecting the relevant contextual information in a movie-recommender system, Interacting with Computers 25 (1) (2013) 74–90.

[87] M. Unger, A. Bar, B. Shapira, L. Rokach, Towards latent context-aware recommendation systems, Knowledge-Based Systems 104 (2016) 165–178.

[88] P. Do, H. Le, V. T. Nguyen, T. N. Dung, A context-aware collaborative filtering algorithm through identifying similar preference trends in different contextual information, in: Advanced in Computer Science and its Applications, Springer, 2014, pp. 339–344.

[89] K. Ji, R. Sun, X. Li, W. Shu, Improving matrix approximation for recommendation via a clustering-based reconstructive method, Neurocomputing 173 (2016) 912–920.

[90] M. Zhang, J. Tang, X. Zhang, X. Xue, Addressing cold start in recommender systems: A semi-supervised co-training algorithm, in: Proc.

37th ACM SIGIR Int. Conf. on Research & development in information retrieval, ACM, 2014, pp. 73–82.

[91] H. Ma, D. Zhou, C. Liu, M. R. Lyu, I. King, Recommender systems with social regularization, in: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, 2011, pp. 287–296.

[92] X. Liu, K. Aberer, Soco: a social network aided context-aware recommender system, in: Proceedings of the 22nd international conference on World Wide Web, ACM, 2013, pp. 781–802.

[93] J. Liu, C. Wu, W. Liu, Bayesian probabilistic matrix factorization with social relations and item contents for recommendation, Decision Support Systems 55 (3) (2013) 838 – 850.

[94] P. Sitkrongwong, S. Maneeroj, P. Samatthiyadikun, A. Takasu, Bayesian probabilistic model for context-aware recommendations, in: Proc. 17th Int. Conf. on Information Integration and Web-based Applications & Services, ACM, 2015, p. 22.

[95] Z. Xu, L. Chen, G. Chen, Topic based context-aware travel recommendation method exploiting geotagged photos, Neurocomputing 155 (2015) 99–107.

[96] X. Ramirez-Garcia, M. García-Valdez, Post-filtering for a restaurant context-aware recommender system, in: Recent Advances on Hybrid Approaches for Designing Intelligent Systems, Springer, 2014, pp. 695–707.

[97] L. Cui, W. Huang, Q. Yan, F. R. Yu, Z. Wen, N. Lu, A novel context-aware recommendation algorithm with two-level svd in social networks, Future Generation Computer Systems.

[98] Y. Zheng, B. Mobasher, R. Burke, Deviation-based contextual slim recommenders, in: Proc. 23rd ACM Int. Conf. on Information and Knowledge Management, ACM, 2014, pp. 271–280.

[99] G. Chen, L. Chen, Augmenting service recommender systems by incorporating contextual opinions from user reviews, User Modeling and User-Adapted Interaction 25 (3) (2015) 295–329.

[100] D. Yang, D. Zhang, Z. Yu, Z. Wang, A sentiment-enhanced personalized location recommendation system, in: Proceedings of the 24th ACM Conference on Hypertext and Social Media, HT '13, ACM, New York, NY, USA, 2013, pp. 119–128.

[101] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu, S. Yang, Social recommendation with cross-domain transferable knowledge, IEEE Trans. Knowl. Data Eng. 27 (11) (2015) 3084–3097.

[102] Y. Hu, Q. Peng, X. Hu, A time-aware and data sparsity tolerant approach for web service recommendation, in: 2014 IEEE International Conference on Web Services, ICWS, 2014, 2014, pp. 33–40.

[103] W. X. Zhao, S. Li, Y. He, E. Y. Chang, J.-R. Wen, X. Li, Connecting social media to e-commerce: Cold-start product recommendation using microblogging information, IEEE Trans. on Knowl. and Data Eng. 28 (5) (2016) 1147–1159.

[104] A. J. Chaney, D. M. Blei, T. Eliassi-Rad, A probabilistic model for using social networks in personalized item recommendation, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, ACM, New York, NY, USA, 2015, pp. 43–50.

[105] J. Li, C. Chen, H. Chen, C. Tong, Towards context-aware social recommendation via individual trust, Knowledge-Based Systems 127 (Supplement C) (2017) 58 – 66.

[106] X. Ren, M. Song, H. E, J. Song, Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation, Neurocomputing 241 (Supplement C) (2017) 38 – 55.

[107] I. Fernández-Tobías, P. G. Campos, I. Cantador, F. Díez, A contextual modeling approach for model-based recommender systems, in: Advances in Artificial Intelligence, Springer, 2013, pp. 42–51.

[108] S. V. Rodríguez, H. L. Viktor, A personalized location aware multicriteria recommender system based on context-aware user preference models, in: Artificial Intelligence Applications and Innovations, Springer, 2013, pp. 30–39.

[109] C. Lin, R. Xie, X. Guan, L. Li, T. Li, Personalized news recommendation via implicit social experts, Information Sciences 254 (2014) 1–18.

[110] A. Q. de Macedo, L. B. Marinho, R. L. T. Santos, Context-aware event recommendation in event-based social networks, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, 2015, pp. 123–130.

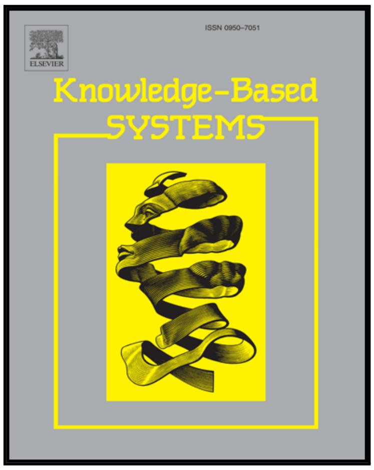
[111] G. Shani, A. Gunawardana, Evaluating recommendation systems, in: Recommender systems handbook, Springer, 2011, pp. 257–297.

[112] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems (TOIS) 22 (1) (2004) 5–53.

[113] R. de Lemos, H. Giese, H. A. Müller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N. M. Villegas, T. Vogel, D. Weyns, L. Baresi, B. Becker, N. Bencomo, Y. Brun, B. Cikic, R. Desmarais, S. Dustdar, G. Engels, K. Geihs, K. M. Göschka, A. Gorla, V. Grassi, P. Inverardi, G. Karsai, J. Kramer, A. Lopes, J. Magee, S. Malek, S. Mankovskii, R. Mirandola, J. Mylopoulos, O. Nierstrasz, M. Pezzè, C. Prehofer, W. Schäfer, R. Schlichting, D. B. Smith, J. P. Sousa, L. Tahvildari, K. Wong, J. Wuttke, Software Engineering for Self-Adaptive Systems: A second Research Roadmap, Vol. 7475, Springer, 2013, pp. 1–32.

[114] M. Litoiu, M. Shaw, G. Tamura, N. M. Villegas, H. A. Müller, H. Giese, R. Rouvoy, E. Rutten, What can control theory teach us about assurances in self-adaptive software systems?, in: R. de Lemos, D. Garlan, C. Ghezzi, H. Giese (Eds.), Software Engineering for Self-Adaptive Systems III, Vol. 9640 of Lecture Notes in Computer Science (LNCS), Springer Berlin Heidelberg, Berlin, Heidelberg, 2017, p. 45, (to appear).

Accepted Manuscript



Characterizing Context-Aware Recommender Systems: A Systematic Literature Review

Norha M. Villegas, Cristian Sánchez, Javier Díaz-Cely, Gabriel Tamura

PII: S0950-7051(17)30507-5
DOI: 10.1016/j.knosys.2017.11.003
Reference: KNOSYS 4098

To appear in: *Knowledge-Based Systems*

Received date: 14 March 2017
Revised date: 31 October 2017
Accepted date: 2 November 2017

Please cite this article as: Norha M. Villegas, Cristian Sánchez, Javier Díaz-Cely, Gabriel Tamura, Characterizing Context-Aware Recommender Systems: A Systematic Literature Review, *Knowledge-Based Systems* (2017), doi: 10.1016/j.knosys.2017.11.003

Characterizing Context-Aware Recommender Systems: A Systematic Literature Review

Norha M. Villegas[a,*], Cristian Sánchez[a], Javier Díaz-Cely[a], Gabriel Tamura[a]

[a] *Universidad Icesi, Calle 18 No. 122-135, 760031, Cali, Colombia*

Abstract

Context-aware recommender systems leverage the value of recommendations by exploiting context information that affects user preferences and situations, with the goal of recommending items that are really relevant to changing user needs. Despite the importance of context-awareness in the recommender systems realm, researchers and practitioners lack guides that help them understand the state of the art and how to exploit context information to smarten up recommender systems. This paper presents the results of a comprehensive systematic literature review we conducted to survey context-aware recommenders and their mechanisms to exploit context information. The main contribution of this paper is a framework that characterizes context-aware recommendation processes in terms of: i) the recommendation techniques used at every stage of the process, ii) the techniques used to incorporate context, and iii) the stages of the process where context is integrated into the system. This systematic literature review provides a clear understanding about the integration of context into recommender systems, including context types more frequently used in the different application domains and validation mechanisms—explained in terms of the used datasets, properties, metrics, and evaluation protocols. The paper concludes with a set of research opportunities in this field.

Keywords: Recommender systems, Context-aware recommender systems, Pre-filtering, Post-filtering, Context modeling, Recommender systems evaluation

*Corresponding author

Email addresses: `nvillega@icesi.edu.co` (Norha M. Villegas), `cesanchez@icesi.edu.co` (Cristian Sánchez), `jgdiaz@icesi.edu.co` (Javier Díaz-Cely), `gtamura@icesi.edu.co` (Gabriel Tamura)

1. Introduction

With the proliferation of big data & data analytics technologies, recommender systems (RS) are now crucial in seeking customer satisfaction through personalization [1]. RS aim at selecting and proposing the most relevant items, services and offers for their users, by considering their profiles, purchase history, preferences, opinions, interactions with offered products and services, as well as their relationships with other clients. At the same time, the generalization of smart-phones and ubiquitous computing has given RS access to context information [2]. Context-aware recommender systems (CARS) go one step further from traditional RS by exploiting context information such as time, location, and user activity to understand user situations and their influence on user preferences. The incorporation of context information into RS [2, 3] leverages the value of these systems by improving the relevance of possible recommendations with respect to changing user needs [4, 5].

The value of context information to improve the quality of recommendations has been demonstrated and supported by different researchers [6, 7, 8, 9, 10, 11]. Nevertheless, RS as well as context-awareness researchers and practitioners interested in combining the two areas still lack a guide that helps them understand how to exploit context information to smarten up RS. Evidence of this is the absence of comprehensive and domain-independent surveys, particularly systematic literature reviews, that not only consolidate the state of the art of the field, but also explain the most common techniques used to integrate context into the recommendation process. After a rigorous revision of the state of the art, we found that none of the available surveys comprehensively characterize recommendation processes from the perspective of the exploitation of context information. In the best cases, existing surveys focus only on the identification of used context types, and most of them address the problem from the perspective of a particular domain.

This paper presents the findings of a systematic literature review (SLR) [12] on CARS that we conducted with the goal of helping practitioners and researchers understand how context information can be effectively combined with recommendation mechanisms. To this end, we studied a final set of 87 CARS papers that were classified as content-based, collaborative filtering and hybrid approaches. For each paper, we identified recommendation techniques, means to exploit context information, context types, application domains, validation mechanisms including the used datasets, the improvements obtained through the exploitation of context (when measured quantitatively), and research opportunities. The main results of our study are reported in this paper in the form of a framework that characterizes recommendation processes in terms of: i) the recommendation techniques used at every stage of the process, ii) the techniques used to incorporate context, and iii) the stages of the process where context is integrated into the system. This manuscript aims at providing a clear understanding about where context information is usually integrated into the system, what techniques are available to exploit context information depending on the underlying recommendation approach and the phase of the process where context is included, what context types are more frequently exploited in the different application domains, and what validation mechanisms—explained in terms of the used datasets, properties, metrics and evaluation protocols—are generally used to evaluate the proposed approaches. Last, but not least, the paper discusses research opportunities relevant to CARS.

This paper is structured as follows. Sect. 2 explains foundational concepts on recommender systems and context information. Sect. 3 visits related work by analysing the contributions of our SLR with respect to other surveys published on CARS. Sect. 4 explains the methodology we followed to conduct the SLR. Sects. 5–8 constitute the contributions of this manuscript: Sect. 5 presents the findings of our SLR and the characterization framework for CARS; Sect. 6 reports on the validation methods and datasets identified in the studied approaches; Sect. 7 presents quantitative data, reported in the studied papers, on the improvements obtained from the exploitation of context information; and Sect. 8 summarizes and classifies research opportunities. Finally, Sect. 9 concludes the paper.

2. Background

This section briefly presents the fundamentals of RS, and context information as an enabler to improve the quality of recommendations.

2.1. Recommender systems (RS)

Dating back to the mid 1990s, the first recommender systems emerged by following two well differentiated paths. On the one hand, *content-based recommenders* drew from the fields of document retrieval [13, 14] and user profiling [15] to define a common representation space for describing items and users. User profiles result from the aggregation of items that have been favorably or unfavorably qualified in the past. For a given user, items similar to the user's profile are recommended, without taking into account information from other users. On the other hand, *collaborative filtering recommenders* evolved from contributions in human computer interaction [16, 17], where the preferences and choices of similar users are used as the basis for recommendation.

Each of these two types of systems has advantages and disadvantages. Content-based recommenders are easy to explain and understand, prove a good starting point for item navigation, and allow recommendations for new users and/or items (cold start problem). However, they imply the cumbersome task of thoroughly and explicitly describing all items using a common set of features, do not work with implied content, can only handle complementary item recommendation and, being centered on a single user, do not allow the recommendation of serendipitous items. In contrast, collaborative filtering recommenders are based on the common preferences of crowds of users. Thus, these systems cannot only recommend complementary as well as substitute items, but also surprise users by recommending unusual items. Nevertheless, they are not as transparent on their recommendations, need substantially more user data to work well, and do not provide a way to deal with the cold start problem.

Hybrid recommenders, a third type of RS, provide a middle ground between content-based and collaborative filtering systems, by leveraging their strengths and mitigating their drawbacks.

This categorization of RS was proposed by Adomavicius and Tuzhilin in [6]. Other authors have proposed other types of systems [1, 18]. In particular, we consider case-based and knowledge-based systems to be subtypes of the content-based family, community-based systems to be subtypes of the collaborative filtering family, and demographic recommenders to be either content-based or collaborative filtering systems following a pre-filtering stage where data are partitioned in subsets according to user characteristics.

RS use information from items, users, and preferences. The main source of information is the item by user matrix that stores user preferences for individual items. These preferences can be explicitly stated (e.g., in the form of ratings or likes), or implicitly inferred from the interactions of the user with the system (e.g., purchases, accesses or reads). Content-based recommenders consider additional sources of information in the form of feature vectors describing different characteristics of each item (e.g., category, size, age, brand, author).

The characterization of CARS presented in this paper is driven by the stages of the processes followed by content-based and collaborative filtering systems.

2.2. Context information

Abowd et al. define *context* as "*any information useful to characterize the situation of an entity (e.g., a user or an item) that can affect the way users interact with systems*"[2]. The precision of recommendations may result highly affected by context information [7, 8]. For example, a costumer could be more or less interested in a particular restaurant depending on the day of the week. Contextual information can be defined as static or dynamic [3]. When context is static, recommender applications assume that this information is immutable over time. An example of static context is the birthday of a user. On the contrary, dynamic context changes over time thus highly affecting user current needs. Instances of dynamic context are location, time, and user activity [5].

2.2.1. Context categories

Villegas et al. [5] characterize context along five general categories: individual, location, time, activity, and relational. Other characterizations, which can be instantiated from these general categories, have been proposed for domain specific CARS (e.g., the one proposed by Verbert et al. in [19] for CARS in the learning realm). To identify the context types exploited by the CARS studied in this SLR, we based on the classification of context information proposed by Villegas et al., which is summarized as follows:

- **Individual context:** Corresponds to information observed from independent entities (e.g., users or items) that may share common features. This category can be sub-classified into *natural*, *human*, *artificial*, or *groups of entities*. *Natural context* represents characteristics of living and non-living entities that occur naturally, that is, without human intervention (e.g. weather information). *Human context* describes user behavior and preferences (e.g., user payment preferences). *Artificial context* describes entities that result from human actions or technical processes (e.g., hardware and software configurations used in e-commerce platforms). The last subcategory, *groups of entities*, concerns groups of independent subjects that share common features, and that might relate each other (e.g., preferences of users in the user's social network).

- **Location context:** Refers to the place associated with an entity's activity (e.g., the city where a user lives). This category is sub-classified as *physical* (e.g., the coordinates of the user's location, a movie theater's address, or the directions to reach the movie theater from the costumer's current location), and *virtual* (e.g., the IP address of a computer that is located within a network).

- **Time context:** Corresponds to information such as time of the day, current time, day of the week, and season of the year. Time context can be categorized as *definite* and *indefinite*. *Definite* context indicates time frames with specific begin and end points. *Indefinite* context refers to recurrent events that occur while another situation takes place, so it does not have a defined duration (e.g. a user's session in an e-commerce application).

- **Activity context:** Refers to the tasks performed by entities (e.g., shopping, the task a user does at a particular time).

- **Relational context:** Refers to entity relationships that arise from the circumstances in which the entities are involved [20]. Relational context can be defined as *social* (i.e., interpersonal relations such as associations or affiliations), *functional* (i.e. the usage than an entity makes of another).

2.2.2. Integrating Context into Recommender Systems

Traditional recommender systems rely on information about users and items. In contrast, CARS rely also on context information that is relevant for the recommendation. Therefore, recommendation tasks in context-aware recommender systems can be seen as a function of users, items and context information [8]:

$$f : Users \times Items \times Context \rightarrow R \tag{1}$$

There exists three paradigms to integrate context information into recommender systems, depending on the phase of the recommendation process at which context is processed [8]:

- **Contextual pre-filtering:** Context information is used as a filtering mechanism applied to the data, before the application of the recommendation model.

- **Contextual post-filtering:** Context information is initially ignored, and preferences are computed by applying traditional recommender algorithms on the entire data. The resulting set of recommendations is then filtered according to context information that is relevant to the user.
- **Contextual modeling:** Context information is directly integrated into the recommendation model, for example as part of the preference computation process.

This SLR characterizes CARS by considering these three paradigms to incorporate context into the recommendation process, and the techniques used for this integration.

3. Related work

We found 15 RS surveys published in relevant venues and journals between 2004 and 2016. However, only 7 out of these 15 surveys, published between 2012 and 2014, relate to the improvement of RS through the incorporation of context information. Aiming at providing a comprehensive understanding of the state of the art of this field, our SLR not only follows a well defined research methodology, but also characterizes CARS along all application domains, context types, and techniques reported in the studied literature. Most importantly, we documented the recommendation processes followed by content-based and collaborative filtering CARS, to characterize how these systems exploit context information along all phases of the process. The characterization includes recommendation techniques, paradigms for incorporating context, context types, application domains, and a detailed explanation of the mechanisms used to exploit context. We also compiled a catalog of datasets and validation methods used in the studied approaches, as well as a list of open challenges.

Table 1 compares our literature review (last row) with the most relevant CARS surveys we found in the state of the art. This comparison is based on seven criteria that we define as follows: *i) SLR*, the literature review follows a systematic methodology; *ii) not focused on particular domains or techniques*, the survey reviews the state of the art across all identified domains and techniques; *iii) not focused on particular context types*, the survey reports the exploitation of different context types; *iv) identifies context exploitation techniques*, the survey reports the ways how context was exploited in the studied RS; *v) context in the stages of the recommendation process*, the literature review documents how context is exploited along the stages of the recommendation process; *vi) datasets*, the survey lists the datasets used by the studied systems; and *vii) validation techiques*, the review reports the techniques used to evaluate the studied approaches. The plus sign in a cell indicates that the survey is compliant with the corresponding criterion, whereas the absence of the sign indicates that it is not.

Table 1: Related work—Comparing our SLR with other surveys on CARS

Author/Year	**SLR**	**Not focused on particular domains or techniques**	**Not focused on particular context types**	**Identifies context exploitation techniques**	**Context in the stages of the recommendation process**	**Datasets**	**Validation techniques**
Verbert et al., 2012 [19]			+	+			+
Kaminskas and Ricci, 2012 [21]			+	+			+
Liu et al., 2013 [22]			+				
Champiri et al, 2014 [23]			+	+			
Campos et al., 2014 [24]		+					+
Inzunza et al., 2016 [25]	+	+	+				
Seifu and Mogalla., 2016 [26]		+	+				
Our literature review	+	+	+	+	+	+	+

According to Table 1, four surveys focus on particular domains: learning processes [19], music services [21], digital libraries [23], and mobile applications [22]. All surveys identify the different types of context exploited in the studied RS, except the one by Campos et al. [24] that focuses on time context only. Furthermore, this survey does not provide insights on the exploitation of context into RS (context exploitation techniques are not identified), but on the evaluation methods used to evaluate the effectiveness of CARS. The surveys conducted by Verbert et al. [19], and Kaminskas and Ricci [21] describe the techniques used to exploit context in the studied systems and the means used to validate them. However, they focus on particular domains. The survey by Liu et al. [22] focuses only on methods to identify the relevant context and the context types exploited in mobile systems. Thus, besides being domain specific, it does not report on techniques used to take advantage of context. As our literature review, the survey conducted by Inzuza et al. [25] follows a systematic approach and does not relate to a particular application domain, technique or context type. However, it does not report on context exploitation techniques. Also similarly to our work, the work conducted by Seifu and Mogalla [26] aims at characterizing the process followed by CARS in the form of what they call *"a framework of CARS."* Nevertheless, their focus is not the way how context is incorporated and exploited, and the explanation of the framework in their six page paper is not as comprehensive as our characterization. Finally, none of the studied surveys report on the used datasets or relate context and its means to exploit it to the concrete phases of the recommendation process.

4. Methodological aspects

We conducted this study by following the guidelines proposed by Kitchenham and Charters in [12]. With our long-term research goal in mind—*to look for innovative and more effective ways of exploiting context information to improve the effectiveness of recommender systems*, we defined the set of research questions that would allow us to understand the state of the art of CARS. These questions are stated as follows:

- RQ1: How is context information exploited along the recommendation process?
- RQ2: What are the existing techniques used to incorporate context information into RS? For each technique, what are the most common application domains?
- RQ3: Is there any correlation between techniques used to incorporate context into RS and any of the traditional recommendation approaches (i.e., content-based, collaborative filtering and hybrid)?
- RQ4: What are the types of context more commonly exploited by RS? What techniques apply in each case?
- RQ5: What evaluation methods have been used to validate the effectiveness of CARS? What are the most common metrics used by these methods?

To answer these research questions and understand the way how context information is integrated into recommender systems, it was important first to characterize the processes that are followed by these systems, in particular by content-based and collaborative filtering approaches. That is, to understand the data that constitute the inputs, and the stages implemented by each type of recommender system to generate recommendations. This process-oriented characterization allowed us not only to report the techniques and context used by the studied RS, but also to map them to specific phases of the recommendation process, with the goal of leveraging the usefulness of this SLR for understanding the state of the art of this field.

We conducted a bibliographic search of conference proceedings and journal papers published in IEEE, ACM, ScienceDirect, EBSCO and Springer. These databases were selected because of the quality of their publications, and their relevance to RS. We used the search string *((”recommendation systems” OR ”recommender systems” OR ”recommendation” OR ”recommendations”) AND (”context aware” OR ”context-aware” OR ”context information” OR ”contextual information” OR ”location” OR ”social” OR ”time” OR ”activity” OR ”task” OR ”environmental”))*.

To select the papers to be included in the study we applied four filters: i) *publication date*, we selected papers published between 2004 and 2016; ii) *publication type, number of citations and language*, we excluded workshop and symposium proceedings, papers with less than 10 citations (with some exceptions for papers recently published) and non-English papers; iii) *relevance*, we studied the abstracts to verify the relevance of each paper. After this third filter, we obtained a total of 286 articles, including surveys on RS.

We thoroughly analyzed all these 286 articles and characterized those proposing CARS according to seven criteria: i) *recommendation system approach*, whether it is content-based, collaborative filtering, or hybrid; ii) *recommendation techniques*, the mechanisms used at the different stages of the recommendation process; iii) *paradigm for incorporating context*, whether it is pre-filtering, post-filtering, or contextual modeling; iv) *context types*, the context categories that are exploited in the recommender system (based on the classification proposed by Villegas and Müller [5]); v) *application domain* (if applicable), the specific area targeted by the proposed RS; vi) *evaluation*, the methods and metrics used to validate the effectiveness of the proposed RS; and vii) *data sets* (when reported), the data used to evaluate the proposed approach.

The fourth and last filter consisted in excluding those papers for which we could not identify any of the mandatory criteria presented above. The final set of papers includes 87 manuscripts that propose CARS and 15 surveys, including four highly relevant papers that were published in 2017.

5. Characterization of Context-Aware RS (CARS)

This section summarizes, for each type of recommender system, the findings of our SLR. We consider that the differences between content-based, collaborative filtering, and hybrid recommenders are too profound to analyze them all together, thus we set to do it independently.

To characterize content-based and collaborative filtering CARS, we first represented their recommendation processes using flow diagrams (cf. Figs. 1 and 2) that allow us to distinguish the different phases they comprise, and identify the points where context information is exploited by the surveyed RS, following either the pre-filtering, post-filtering or contextual modeling paradigms.

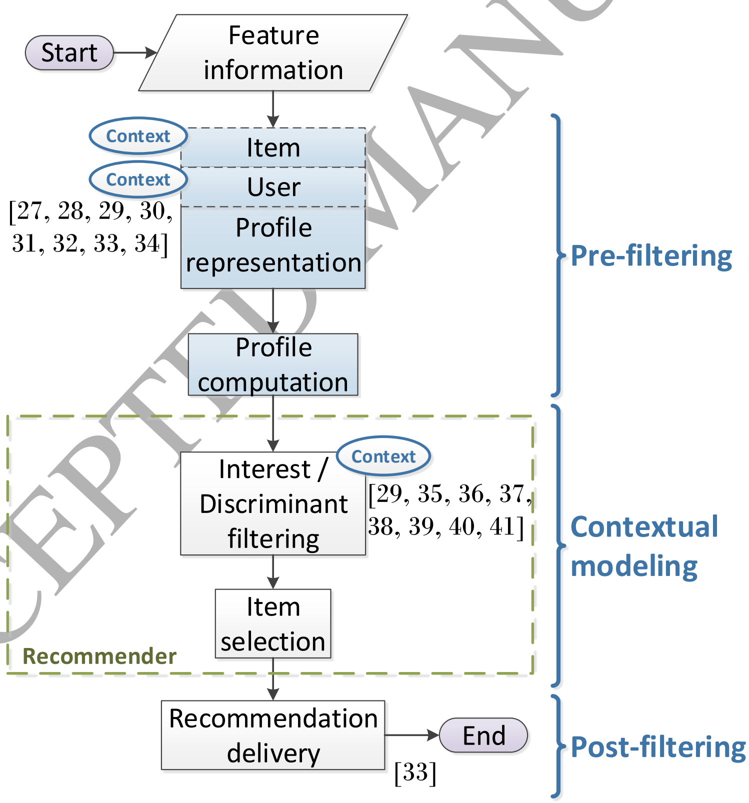
Bold ovals labeled as “Context” indicate the points of the process where we consider that context information can be incorporated. Citations next to each oval correspond to the studied approaches that integrate context in that specific phase of the recommendation process. The absence of citations next to an oval indicates that although we consider context can be exploited at that phase of the process, we found no approaches that do so. Furthermore, each brace depicted in the diagrams groups the stages of the process associated with each of the three paradigms commonly used to incorporate context into a CARS (i.e., pre-filtering, contextual modeling and post-filtering).

It is important to stress out that we focus on the ways in which context can be incorporated and exploited in the recommendation process. Even though on the diagrams we illustrate that process as a whole, we are mainly interested in showing the specific points where the reviewed papers (their references are placed accordingly on the diagrams) decided to adapt the recommendation process to exploit context. While we rely on some of the reviewed papers to illustrate the characterization and findings presented in this section, a more detailed retelling on how each paper implements their system and incorporates context can be found on Tables 2 and 3.

5.1. Content-based approaches

We found 15 papers associated with content-based CARS. Table 2 summarizes the characterization of these papers (cf. Column *Appr.*), which is driven by the process depicted in Fig. 1. Columns *Profile representation*, *Profile computation*, and *Discr. filter* indicate the techniques implemented by the studied systems to realize the main phases of the content-based recommendation process. Column *Paradigm* denotes the strategy used to incorporate context information: pre-filtering, contextual modeling, post-filtering. Column *Context Types* corresponds to the context categories exploited by the corresponding approach. Column *Domains* lists the application domains for which the RS was proposed. The last column explains the means used by the studied CARS to exploit context information.

Figure 1: Process followed by content-based CARS



5.1.1. The beginning of the process

The process implemented by content-based CARS (cf. Fig. 1) begins with the identification of the features in the available data that will define the common dimensional space used to describe item characteristics and user preferences (cf. *User profile definition* and *Item profile definition* in Fig. 1).

Pre-filtering strategies are applicable through the incorporation of contextual factors in the definition of item and/or user profiles. These strategies reduce significantly the search space for the discriminant filter by initially discarding a part of the information available. However, they require the inclusion of redundant user or item profiles for different contextual situations.

All content-based reviewed papers defined the features used as the basis for their recommendation, but only about half of them included contextual information as features. CARS proposed in [27, 28, 29, 30, 31, 32, 33, 34] exploit context using a pre-filtering strategy to generate different contextual *user* profiles for the same user, with different preferences for different situations (see Table 2 for more details regarding the four papers that apply pre-filtering as the paradigm to incorporate context). For instance, [29] proposes a movie CARS where contextual variables of different types such as time (weekday, weekend), location (theater, home), and social context (companion, friends, family) are taken into account to consider or ignore past user ratings, by building several context-aware (micro) profiles that are used to generate context-aware recommendations. As a result, the same user can have different profiles.

None of the surveyed papers associate contextual information with items. We assume that this is because it is easier to think in terms of contextual user profiles than in terms of contextual item profiles, probably because user preferences naturally vary according to context situations. Still, it is completely possible to have different *item* profiles for different situations. Nevertheless, since very often the number of items is many times larger than the number of users, it would mean increasing the complexity of the recommendation process given that a considerably larger number of items must be handled by the system.

5.1.2. The core of the process

The next phase is the core of the recommendation process. In general, a discriminant filter working as a utility function between user and item profiles is responsible for generating a recommendation score from the item and user vectors. This can be done through several strategies: i) by applying some similarity measure such as *Cosine Similarity* (since items and users are represented on the same dimensional feature space, it is possible to compute the distances or similarities between them, with the goal of selecting the items closer to the user's preferences [27, 28, 29, 30, 31, 34, 35, 36]); ii) by obtaining a given classification score by applying a supervised learning technique ([37, 38, 39]); or iii) by applying a heuristic approach (context information can be considered into a discriminant filter, not as additional profile dimensions, but as an integral part of the function definition [32, 33, 35, 40, 41]). Either way, the recommender engine will associate a numeric value to each item, order the items accordingly, and select the ones that appear at the top or that surpass a specified threshold.

At this stage of the process, contextual information can be incorporated by influencing the similarity or distance between items and users. For example, [36] proposes a music recommendation system that incorporates the time at which users accessed different items (songs) in order to provide more relevant recommendations. In their system, users are described by a vector of their correlations to the considered time-related contexts (dawn, morning, monday, tuesday, spring, christmas), items are described as a vector of their correlations to the domain features (e.g., band, genre) as computed by a TF-IDF measure, and the historical accesses to items by users are kept as a collection of pairs of vectors as previously described. To perform a recommendation, the cosine similarity measure is applied to the user's current context and the historical accesses, the similarity of the historical accesses to the items is computed, and an aggregation of both measures allows the scoring of every available songs, so that the top five songs are presented to the user.

5.1.3. The end of the process

Finally, the selected recommendations are organized and delivered to the user. Post-filtering strategies apply at this stage to eliminate the recommendations that are irrelevant to the user's current context. We found that only the RS presented in [33] applied this paradigm to filter out movie recommendations that did not correspond to the current time and location.

5.1.4. Findings

Regarding the paradigm used to incorporate context into the RS (cf. Sect. 2.2.2), findings show that content-based approaches use contextual modeling as much as pre-filtering (one of those combining both strategies);

both paradigms being followed by 53% of the papers. Only one of the studied content-based CARS [33] incorporates context information using post-filtering, combined with pre-filtering. We hypothesize that this may be in part because post-filtering strategies may result in wasting time and computational resources, since the obtained recommendations may become useless after evaluating them with respect to the current context of the user, which is taken into account only at the end of the process. Indeed, pre-filtering approaches provide more benefits in what respects to computational complexity, and contextual-modeling solutions have proven to be more effective for the accuracy of recommendations [4].

With respect to the types of contextual information commonly used in the reviewed systems, and their application domains, we found that time context is commonly used in application domains such as movies and news; location context, in domains associated with movies, music and points of interest; activity context in domains related to movies, music and points of interest; social context in multimedia applications and human context in web services recommendations. It is of particular interest that none of the reviewed content-based CARS target the e-retailing domain, an otherwise popular application domain in traditional RS.

Table 2: Characterization of content-based approaches

Appr.	Profile representation	Profile computation	Discr. filter	Paradigm	Context types	Domains	Means to incorporate context
[28]	Item features	Case based reasoning (CBR)	Cosine similarity	Pre-filtering	Time, Location, Activity, Artificial (environment)	Movies, Music, News	Generates a contextual user profile by revising the user's consumption behavior. Then, it uses cosine correlation to measure the similarity between the user contextual profile and the item profile.
[37]	Item features	Heuristic approach	Decision tree algorithm	Cont. Model.	Activity, Human (age, gender)	Indoor Shopping, Points of interest	Proposes a framework where the relationship between user profiles and services under the same context situation are analyzed to infer user preference rules, using the decision tree algorithm.
[27]	Item features structured by a reference ontology	Heuristic approach	Cosine similarity	Pre-filtering	Activity, Time, Location	Movies	Tracks user browsing behavior, and understands user preferences in each particular context. Then, it performs recommendations by means of an aggregation agent that selects the top N items with the highest inferred values.

Table 2: Characterization of content-based approaches

Appr.	Profile representation	Profile computation	Discr. filter	Paradigm	Context types	Domains	Means to incorporate context
[30]	Tag-based features	Heuristic approach	Cosine similarity	Pre-filtering	Time, Location, Activity, Natural (Weather)	Points of interest	Uses a relational Markov network to match the features of Points of Interest (POI) with the current context. POI's features (e.g. outdoor seating, waiter service, dinner) are taken as the inputs to a neural network used to classify the appropriate level of interest (5 categories) of the user for the POI, under the given context situation. The resulting vector that characterizes the POI is then compared to the user vector using cosine similarity.
[29]	Item features	Heuristic approach	Cosine similarity	Pre-filtering, Cont. Model.	Time, Social, Location	Movies	Pre-filtering: Splits user ratings according to the contextual situation in which the preference is expressed, then builds several context-aware (micro) profiles used to infer preferences for new products. Contextual Modeling: Considers context as a weighting factor that influences the recommendation score of a user for a certain item. It combines the non-contextual vector space representation of user preferences with a vector space representation of context, which is built using the pre-filtering approach.
[35]	Latent semantic features	Term frequency inverse document frequency (TF-IDF)	Cosine similarity	Cont. Model.	Location	News	User is defined by the articles read in the past along with his/her location. The system seeks to rank a set of articles that satisfy the geographical location of the user. The preference score is determined by a cosine function ($f(a, l)$) that measures the appropriateness of each article a to a location l.
[38]	Item features	Heuristic approach	Joint probabilistic distribution	Cont. Model.	Activity	Music	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. The first part computes a probability distribution using the Bayes' rule. The second part computes a prior probability that captures the history of user preferences.
[40]	Item features	Heuristic approach	Heuristic approach	Cont. Model.	Location	Indoor shopping	Focuses on mobile recommender systems for assisting indoor shopping by considering location-context. User preferences are calculated through a heuristic approach that integrates three factors: i) time spent in a brand store, ii) frequency of visits to the store, and ii) the matching between the special offers or promotional activities done in the brand store and the user's preferences.
[36]	Item features	Term frequency inverse document frequency (TF-IDF)	Cosine similarity	Cont. Model.	Time	Music	Context refers to the time at which the user listens to a song. The approach predicts user preferences by: i) computing the similarity between the user's current and historical contexts, ii) computing the correlation between historical context and an item, and iii) deriving the expected preference by multiplying measures obtained in i) and ii).

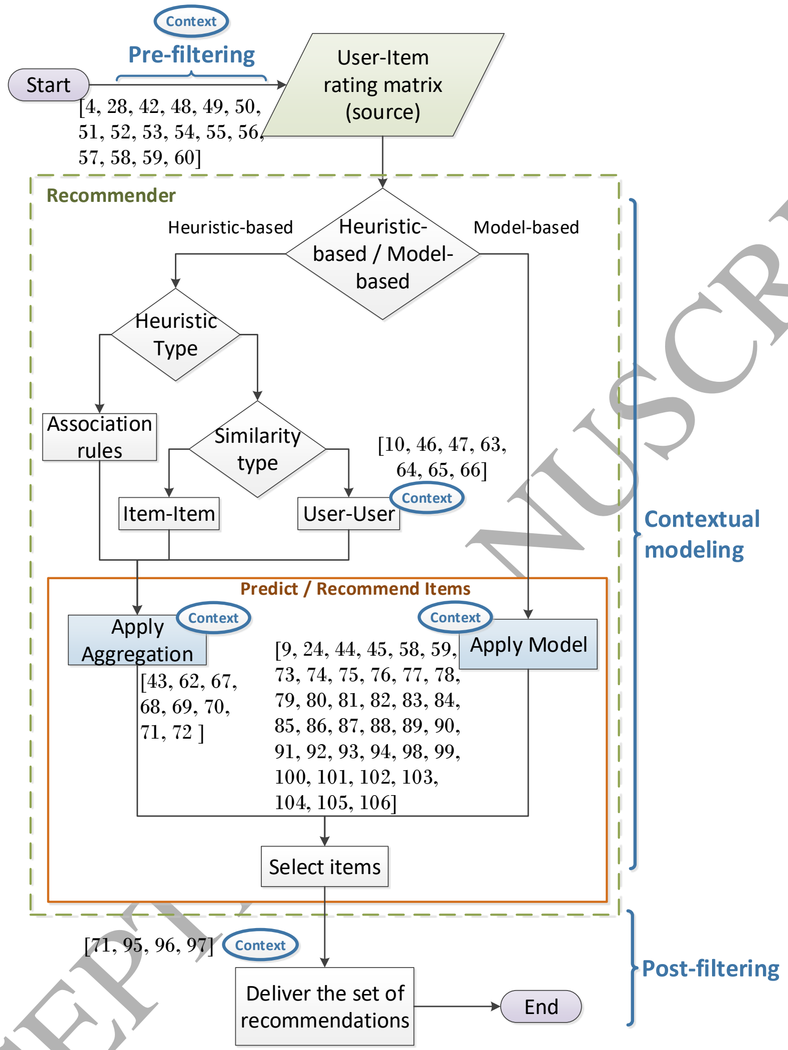
Table 2: Characterization of content-based approaches

Appr.	Profile representation	Profile computation	Discr. filter	Paradigm	Context types	Domains	Means to incorporate context
[39]	Latent semantic features	Heuristic approach	Joint probabilistic distribution	Cont. Model.	Activity, Location	Music	Implements a recommendation model where a set of latent topics is used to associate music content with a user's music preferences under certain location. It is based on the joint probability distribution of user, place, song and lyrics. The latent topics are the intrinsic factors that explain why users prefer certain pieces of music in a particular location and during a specific time period.
[31]	Item features	TF-IDF	Cosine similarity, Jacard similarity	Pre-filtering	Human (user-interest)	Web services	Infers user preferences from the description of the web services that have been accessed by the user.
[32]	Item features	Heuristic	Heuristic	Pre-filtering	Social (followers)	Multimedia	Utilizes Social context (followers) as the basis to decide on user-similarity.
[41]	Item features	Heuristic	Heuristic	Contextual modeling	Location	Points of Interest	Considers context as a weighting factor that influences the recommendation score of a user for a certain item.
[33]	Item property	Heuristic	Heuristic	Pre-Filt, Post-Filt.	Location, Time	Movies	Recommends items with a composite structure (movie theater + movie + showtime). This approach first computes a similarity metric that concerns to the relation between the composite item (theater, movie, showtime) -Pre-Filtering. Then, this similarity measure is incorporated into the discriminant filter -Post-Filtering.
[34]	Item feature	Heuristic	Euclidian Distance	Pre-filtering	Activity	General application	Utilizes a sequential patterns method to find rules from data records on users' smart-phones. Then, by detecting and matching the user's current situation to the rules, which consider his current context and the events in which he has participated, the system determines the most suitable rules for making just-in-time recommendations.

5.2. Collaborative filtering approaches

Figure 2 depicts the general process followed by collaborative filtering CARS. Based on this process, we characterized the 69 collaborative filtering CARS studied in our SLR. This characterization is summarized in Table 3. Column *Recommendation strategy* presents the techniques implemented by the studied approaches, which can follow different paths of the recommendation process, as explained later in this section. As in the characterization of content-based CARS (cf. Table 2), the characterization of collaborative filtering CARS includes the paradigm used to incorporate context into the system (cf. column *Paradigm*), the types of context information exploited by the studied approaches (cf. column *Context Types*), the application domain (cf. column *Domains*) and the mechanisms used to exploit context (cf. column *Means to incorporate context*).

Figure 2: Process followed by collaborative filtering CARS



5.2.1. The beginning of the process

The input of the collaborative filtering process is a user-item rating matrix, where usually rows represent users, and columns represent items. This matrix can include additional dimensions to represent contextual information in the form of synthetic columns or rows, as in the case of the systems presented in [4, 42, 43, 44]. For example, Baltrunas et al. [42] extend the user-item rating matrix into a user-item-context matrix, where contextual information consists in categorical tags (e.g. sunny, cloudy, raining) associated with a given rating.

Depending on the application domain, this matrix can be either obtained directly from the interactions of users with items (e.g., by capturing media accesses instantly [28, 45, 46, 47]), or inferred from historical interactions stored in transactional databases (e.g., by analyzing event logs of previous accesses to the recommended items [10, 48]). This matrix can be very sparse and its processing can be computationally challenging when the number of users and items is considerable (several hundreds of thousands).

At the beginning of the process, *pre-filtering* strategies generate different contextual user-item rating matrices, independent of each other. On the one hand, pre-filtering strategies reduce computational complexity since only a portion of the rating matrix is considered; on the other hand, they imply an extra effort in the acquisition of information, since ratings must be generated for every contextual situation that remains relevant after applying the filter.

We identified 16 papers reporting on the application of context-based pre-filtering strategies to generate recommendations [4, 28, 42, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60]. Pre-filtering is a simple strategy that discards a large part of the data to be analyzed, according to the user's current context. An instance is the process followed by the CARS proposed by Lee et al. [48], in which the authors analyze the access logs to songs, and extract context from the timestamps. Then, they define fuzzy membership functions to fuzzy sets for different contextual variables such as season, time of day, or day of the week, in such a way that the same song recommended at different moments is not considered to be the same item. Another example of collaborative-filtering pre-filtering CARS is the one proposed by Baltrunas et al. [42]: if a statistical test shows that context affects the consumption of an item, they split the item into several synthetic items according to the context situation. For instance, a movie could be split into the same movie associated with winter time, and another one associated with summer time.

5.2.2. The core of the process

To perform the actual recommendation, we identified that most systems apply one of two types of collaborative filtering approaches: *heuristic-based* and *model-based* methods. We found no relationships between any of these methods and particular application domains.

Heuristic-based methods. In the studied systems, heuristic-based approaches are realized through association rules, or the analysis of similarities between users or items. The Apriori algorithm [61] is a common technique for association rule learning. First, it identifies the frequent individual items in the database. Then, it extends them to larger itemsets as long as those appear often enough in the database. Finally, these itemsets are used to determine association rules that allow the discovery of hidden relationships in the data, based on the conditional probability existing between itemsets. The association rules approach is mainly applied to transactional data. However, it can also be applied to the user-item rating matrix, by considering each user row as a single transaction.

An interesting finding of our SLR is that despite approaches such as the one reported in [62] mine association rules, none of the studied systems exploit this technique to incorporate context. A reason for this could be that it would imply extra efforts to acquire the information required to generate a more comprehensive rating matrix, such that the extracted rules are meaningful enough in terms of support, and include context in rule antecedents.

Heuristic-based approaches based on similarity analysis consist in determining the distance between users or items. Each user can be seen as a vector in a feature space with an independent dimension associated with each item (and vice-versa). In general, these distances are determined using neighbourhood or clustering-based methods.

These methods work in two ways. The first one, user-user collaborative filtering, consists in inferring user preferences by determining the group of users that are more similar to the target user, and aggregating the items that are most popular among the members of the user group. The second one, item-item collaborative filtering, consists in determining the similarity among items rated by similar users. In either case, the method requires the computation of the distances between users or items, which can be computationally demanding when dealing with a considerable number of users or items.

Seven of the heuristic-based approaches included in this SLR incorporate context through user-user similarity matching; for instance, the approaches presented in [10, 46, 47, 63, 64, 65, 66] incorporate context to the analysis of user-user similarities (more details can be found on Table 3). On the other hand, none of the heuristic-based approaches use item-item collaborative filtering to incorporate context. As discussed previously, we hypothesize that this is because it results more natural to associate context with users than with items. Nevertheless, in some application domains (e.g., products that are mainly consumed in a particular time of the day), context can be effectively associated with items, in which case an item-item collaborative filtering method that incorporates context would be an appropriate strategy.

Continuing with the recommendation process based on heuristic methods, the information obtained from applying the selected method is aggregated to rank the items to be recommended. Eight of the reviewed papers correspond to collaborative filtering RS that incorporate context as additional factors in the aggregation function. In particular, by using a maximization function [43, 62], a sum of products [67, 68, 69, 70, 71], and probabilities [72]. For instance, Khalid et al. [62] combine the approximated time required to reach a restaurant, the road speed conditions and the distance from the user into a defined metric. Then, the restaurant maximizing this metric is recommended to the user.

Model-based methods. Model-based approaches rely mostly on latent factor models applied to the user-item rating matrix. As we have said before, we can interpret this matrix as either a multi-dimensional representation space where each user is a vector with each item as a dimension, or a multi-dimensional representation space where each item is a vector with each user as a dimension.

The idea of latent factors RS is to obtain a single multi-dimensional space where both users and items can be represented, side by side, through matrix decomposition techniques. In this latent space (usually of smaller dimensionality than the user-item rating space), it is then possible to compute similarities and distances between users and users, users and items, and items and items.

We identified that some systems introduce contextual factors as additional dimensions of the original matrix (e.g., [44, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83]), while some other include contextual information as additive biases on users and items, to affect the calculation of missing ratings (e.g., [9, 45, 84, 85, 86, 87, 88, 89, 90, 91, 92]). An example of the first group is presented in [73], where the authors perform contextual recommendations using tensor factorization. This technique stores the latent feature of users, items and context types in three different matrices. Then, ratings are calculated as the inner product of the latent feature vectors of the given matrices. As a case of the second group, we can consider the RS presented in [85], which performs context-aware recommendations by incorporating temporal changes into the matrix factorization technique. In particular, this approach seeks to capture past temporal patterns over products and items to predict future behaviour, and thus infer preferences. A particular case is the approach presented by Liu et al. [93], which incorporates social context from a social network into the recommendation model by considering that users belonging to different social groups should have different hyperparameters to be used during the matrix factorization process.

It is important to note that despite the collaborative filtering recommendation process indicates that heuristic-based and model-based techniques are not commonly used together, the authors of papers [9] and [88] propose CARS where model-based and heuristic-based techniques are combined. For instance, in [9] user interactions are represented in the form of a social network graph, where each node represents a user, and arc weights correspond to the trust existing between users represented by adjacent nodes (i.e., social context). This approach uses a heuristic-based technique (i.e., graph theory) along with a model-based method (i.e., matrix factorization).

We found a few papers reporting on the application of other approaches. In particular, machine learning techniques, where context information is usually incorporated by implementing probabilistic models such as the Bayesian model [24, 94], or the usage of classifiers such as support vector machines [81, 82, 83].

5.2.3. The end of the process

Similarly to content-based CARS, at the end of the process a contextual filter can be applied to the resulting recommendations to eliminate those items that are irrelevant to the current context. We found four papers reporting on the incorporation of context as a post-filtering strategy to ignore [95], filter [72, 96, 97], or adjust [72, 96] the inferred recommendations.

For example, the systems reported in [72, 96] ignore context until a traditional collaborative filtering algorithm produces restaurant recommendations, which are then adjusted to the user's current context.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[4]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr.:** Sum of products	Pre-filt.	Time, Social, Location	Movies	Filter information according to the current context. A rating is computed for the given user and item, as an aggregation of the ratings of other similar users.
[48]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr.:** Sum of products	Pre-filt.		Music	
[49]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr.:** Top N (most important users)	Pre-filt.		Movies	
[50, 51]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr.:** Sum of products	Pre-filt.	Time, Location	Movies	
[52]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr.:**Sum of products	Pre-filt.	Location	Points of interest	
[28]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr.:**Sum of products	Pre-filt.	Location, Activity, Artificial (environment)	Movies, Music, News	
[53]	**Heuristic-based**, **User & Item sim:** K-medians, **Aggr.:** Maximum	Pre-filt.	Time	E-retailing	
	Heuristic-based, **User & Item sim:** Graph theory, **Aggr.:** Maximum				
[54]	**Heuristic-based**, **User sim:** Graph Theory, **Aggr.:** Maximum	Pre-filt.	Location, Social	Points of Interest	
[60]	**Heuristic-based**, **User & Item sim:** Cosine similarity, **Aggr.:** Sum of products	Pre-filt.	Time	Movies, Music	The authors propose a neighbor-based collaborative filtering approach. A similarity measure over human and time contextual factors provides the basis for estimating the neighborhood of both users and items that will be considered in the recommendation process.
[42]	**Model-based**, **Tech.:** Matrix Fact.	Pre-filt.	Time, Social	Movies	Splits items that have been rated under different context situations. This split is performed only if there is statistical evidence that under these context situations users rate items differently.
[55]	**Model-based**, **Tech.:** Markov Chains	Pre-filt.	Time, Activity	General application	Processes user historical logs to extract contextual features such as day, time range, and location. Then, it identifies common preferences under different contextual conditions. Finally, it makes recommendations based on distributions of user preferences.
[56]	**Heuristic-based**, **User Sim:** Graph theory, **Aggr:** Sum of products	Pre-Filt.	Social	Music, E-retailing	Examines the context-aware recommendation as a search problem in the contextual graph.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[57]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr:** Sum of products	Pre-Filt.	Different types	General Application	Context information associated with users is exploited to infer individual user profiles and from these, the profiles of the groups.
[58]	**Model-based**, **Tech:** Matrix Fact.	Pre-Filt., Cont. Model	Location, Time	Hotels & Tourism	The original user-item rating matrix is divided into sub-matrices according to the temporal states. Then, each sub-matrix is factorized by considering location characteristics.
[59]	**Model based:**, **Tech:** Matrix Fact.	Pre-Filt., Cont. Model.	Location	Web services	Users and services are clustered into groups according to their location. These are then characterized according to their particular QoS features into a local user-service matrix. There is also a global user-service matrix where location is not considered. Matrix factorization is performed on the local and global matrices in a step-wise hierarchical linear process
[46]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr.:** Sum of products	Cont. Model.	Location, Time	Points of interest	Adopts an adjusted Pearson coefficient that computes similarities between users in different contexts. In order to do so, the approach defines a context similarity matrix that includes the coefficient between two users' current contexts for using an item. This coefficient is then incorporated into the aggregation function that computes the missing ratings.
[62]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr.:** Maximum	Cont. Model.	Location, Time	Points of interest	Recommends restaurants by computing the approximate time in reaching it, and considering distance, speed and road conditions. This approximation is included into the aggregation function.
[43]	**Heuristic-based**, **Item sim:** Cosine similarity, **Aggr.:** Maximum	Cont. Model.	Location, Time	Points of interest, Music	Transforms the initial user-item matrix by integrating contextual factors as virtual items.
	Heuristic-based, **As. Rules:** Apriori, **Aggr.:** Maximum				
[67]	**Heuristic-based**, **Item sim:** Pearson correlation / Cosine Similarity, **Aggr.:** Sum of products	Cont. Model.	Human (mood), Time	E-learning	
[10]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr.:** Sum of products	Cont. Model.	Time, Human (intent of purchase: Personal-work, Gift Partner, Friend, Parent)	E-retailing	Considers virtual users under different contexts and finds neighbors of contextually similar users to infer recommendations.
[47]	**Heuristic-based**, **User sim:** Jaccard Similarity, **Aggr.:** Sum of products	Cont. Model.	Location, Time	Points of interest	Modifies the Jaccard similarity measure to incorporate context.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[63]	**Heuristic-based**, **User sim:** Pearson Coefficient, **Aggr.:** Sum of products	Cont. Model.	Social	General application	Integrates the strength of the relationships between telecom users into the similarity measure. This strength is modeled taking into account context information associated with phone calls such as duration, time of day and day of the week.
[9]	**Model-based**, **User sim:** Graph theory, **Tech.:** Matrix Factorization	Cont. Model.	Social	Movies	Combines the user-item rating matrix with user-user social contextual information from a trust network to generate a modified rating matrix. This last matrix is then factorized.
[84]	**Model-based**, **Tech.:** Matrix Factorization	Cont. Model.	Social	General application	
[85]	**Model-based**, **Tech.:** Matrix Factorization	Cont. Model.	Time	Movies	Consider context information to add biases on users and items into the recommendation model. Rating values are then influenced by context changes.
[45]			Time	Points of interest	
[86]			Time, Location	Movies	
[87]			Location, Time, Activity	Points of interest	
[91]			Social	Books, Music, Movies	
[92]			Social	General application	
[88]	**Model-based**, **Tech.:** Matrix Factorization	Cont. Model.	Time, Human (Hunger level, mood)	Food, Movies	Clusters items into groups according to the context of their consumption and treats them as virtual items associated with users in a new matrix that is then factorized. Missing ratings are inferred taken into account contextual information.
[89]	**Model-based**, **Tech.:** Matrix Factorization	Cont. Model.	Social	Books, Music, Movies	Considers context information to add biases on users and items into the recommendation model. Through matrix factorization, it creates a common latent factor space for users and items. In this representation space, users and items are clustered independently, so that they can then be brought back to a user-item rating matrix, where missing ratings can be inferred for groups of users.
[90]	**Model-based**, **Tech.:** Matrix Factorization	Cont. Model.	Human (age, gender)	Movies	Constructs several prediction models based on matrix factorization. Each model is then refined by taking into account the predictions from other models. Context information is considered to add biases on users and items into the recommendation model. Rating values are then influenced by context changes.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[73, 74]	**Model-based**, **Tech.:** Tensor Factorization	Cont. Model.	Time, Human, Social	Movies	Perform context-aware recommendations using tensor factorization, which considers the latent features of users and items, and the interaction of the user with an item under a given context. The latent feature of users, items and context types are stored in three matrices. Thus, the inference of preferences is computed as the inner product of the latent feature vectors of the matrices.
[75]			Time	Movies	
[76]			Location, Activity	E-retailing	
[77]			Social, Time	Movies, Food	
[78]			Social, Time	E-retailing, Movies	
[79]			Human (hunger level), Time, Location	Food	
[80]			Social, Time	E-retailing, Movies	
[81, 82]	**Model-based**, **Tech.:** Support Vector Machine (SVD)	Cont. Model.	Time, Social, Natural (weather), Location	Points of interest	Apply SVD to the ratings as represented in a user-item-context space to discriminate between recommended and not recommend items.
[83]	**Model-based**, **Tech.:** Support Vector Machine (SVD)	Cont. Model.	Location	Points of interest	
[94]	**Model-based**, **Tech.:** Bayesian Model	Cont. Model.	Time, Location, Human (mood)	Movies	By adopting a binary particle-swarm optimization technique, identifies the relevant contextual factors for user and item classes, and incorporates them into a latent probabilistic model.
[24]	**Model-based**, **Tech.:** Naïve Bayes	Cont. Model.	Time	Movies	Identifies which members of a household made some specific unidentified ratings of movies by considering time-context conditions such as hour of the day, day of the week and date of rating, as well as number of ratings given by a user. To do this, it analyses temporal trends using probability models.
[98]	**Model-based**, **Tech.:** Sparse Linear Method	Cont. Model.	Time, Location, Social	Movies	Models the contextual rating deviations of items, by assuming that there is a rating deviation for each <item, context condition> pair. This deviation is represented in a matrix, where each row represents an item, and each column represents an individual contextual condition. Then, the ranking score is estimated by an aggregation of user ratings on other items in the same context.
[99]	**Model-based**, **Tech.:** Linear Regression	Cont. Model.	Social, Time	Hotels & Tourism	Predicts user preferences using a linear regression model, which includes a value that represents the user context preference. This value can be computed by means of three different probabilistic methods: i) mutual information based method, ii) information gain based method, and iii) chi-square statistic based method.

ACCEPTED MANUSCRIPT

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[100]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Location, Social	Points of interest, Hotels & Tourism	Location of venues and user social network information are integrated into the matrix factorization model.
[64]	**Heuristic-based**, **Item & User sim:**Pearson correlation, **Aggr:** Weighted ad-hoc	Cont. Model.	Social	Web services	The level of trust among users (social context) is included in the weighted aggregation
[65]	**Heuristic-based**, **User sim:** Ad hoc, **Aggr:** Ad hoc	Cont. Model.	Location, Social	Points of interest, Hotels & Tourism	The social (relationships) and location context of the user is integrated into the process to measure the similarity between users.
[44]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Time, Activity, Location, Artificial	General application	Context-aware preferences as dimensions of the matrix
[93]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model	Social	E-retailing	Social context is considered in order to define groups of users with particular hyper-parameters used by the matrix factorization model
[68]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr:** Sum of Products	Cont. Model	Time, Social	General application	The prediction of user's preference is affected by the user-similiraty , which is computed by considering the context (i.e, the social taggins)
[69]	**Heuristic-based**, **User sim:** Pearson correlation, **Aggr:** Sum of Products	Cont. Model	Time	Movies	Adds a time dimension to the original input data. It is defined in a new table which shows item ratings for an active user at different time-frames.
[70]	**Heuristic-based**, **User sim:** Cosine similarity, **Aggr:** Sum of products	Cont. Model	Time	Music	Infers user's preference by considering a context score, which is computed for each item in the recommendation list which shows the suitability of that item for the current context of the user.
[101]	**Model-based**, **Tech:** Random walk	Cont. Model.	Social	Social Networks	Tags from social networks are the basis for user similarity (Jaccard). Posts from users are compared by applying an ad-hoc similarity measure. A random walk algorithm is applied in order to estimate weights relating users to users in the social domain and users to items on auxiliary domains (web posts, videos, labels)
[102]	**Model-based**, **Random walk**	Cont. Model.	Time	Web services	Making time-aware personalized QoS prediction is important for high-quality web service recommendation because their performance is highly correlated with invocation time, since service status and network conditions are continuously changing. Time is integrated into a modified Pearson correlation similarity measure (similarities between users and between web services); time is also considered when making the final QoS prediction.
[103]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Social, Time	E-retailing	Social networking features of users (demographics, user posts, groups of related users, temporal activity preferences) that also interact with an unrelated e-commerce site can be transformed into latent factors that can be used for product recommendation, particularly for unknown new users of the e-commerce site.

Table 3: Characterization of collaborative filtering approaches

Appr.	Recommendation strategy	Paradigm	Context types	Domains	Means to incorporate context
[104]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Social	Retailing	The authors propose Social Poisson Factorization (SPF) probabilistic model that incorporates social network information into a traditional factorization method, assuming that each user's clicks are driven by their latent preferences for items and the latent influence of their friends (modeled as conditional probabilities). SPF also allows for generating explanations of recommendations based on the social relationships of users.
[105]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Social	Retailing	A probability based matrix factorization is proposed, taking into account trust relationships in a social network in the item recommendation process for retailing purposes. Users and items are then clustered using a Gaussian Mixture Model to enhance the recommendation performance.
[106]	**Model-based**, **Tech:** Matrix Fact.	Cont. Model.	Location, Social	Points of interest	The authors propose a probabilistic matrix factorization method which considers contextual information taken from a location-based social network, where each point of interest is described using a topic model, geographical and social correlations.
[66]	**Heuristic-based**, **User sim:** Jaccard similarity, **Aggr:** Heuristic graph based.	Cont. Model.	Social	Social Networks	The social features of folksonomies are used to provide a user with recommendations of similar users and resources. User profiles consider social contexts, by incorporating information of actions performed by the user on neighboring users' tags, and of other neighboring users on the user's tags. User neighborhoods are defined based on the social network friend relationships according to a specified length of the minimum path linking two users.
[71]	**Heuristic-based**, **User-Sim:**Ad-hoc, **Aggr:** Sum of products	Cont. Model.	Social	E-retailing	Adopts an ad-hoc similarity measure that computes similarities between users in different social context. This measure is then incorporated into the aggregation function that computes the missing ratings
[72]	**Heuristic-based**, **User sim:** Graph theory, **Aggr.:** Probability	Cont. Model., Post-Filt.	Time, Location, Natural (weather), Social	Movies, Hotels & Tourism	Proposes a graph-based contextual model framework. It examines the context-aware recommendation as a search problem in the contextual graph. It also includes a probabilistic-based post-filtering strategy to improve the recommendation results giving contextual factors.
[97]	**Model-based**, **Tech:** Matrix Fact.	Post-Filt.	Time	Movies	The authors propose two successive SVD matrix factorizations to further refine the latent factors for users and items independently, while using time context to filter out unfit items.
[95]	**Heuristic-based**, **User sim:** Cosine Similarity, **Aggr.:** Sum of products	Post-Filt.	Location, Time, Natural (weather)	Hotels & Tourism	Keeps track of contextual features of past user travels to each location. Context aware recommendations are inferred by finding the most similar users, calculating a score for each location, and filtering locations that do not meet contextual conditions.
[96]	**Heuristic-based**, **Users sim:** Pearson correlation, **Aggr.:** Sum of products	Post-Filt.	Time, Location	Points of interest	Adjusts inferred ratings to deliver contextual recommendations.

5.2.4. Findings

The information summarized in Table 3 suggests a correlation between the strategy used to generate recommendations and the paradigm used to incorporate context into the recommendation process of collaborative-filtering CARS.

In general, model-based approaches incorporate context using contextual modeling. This can be explained by the fact that models provide a more natural way to capture interactions between users, items and context. We also found papers reporting on the combination of model-based methods and pre-filtering strategies [42, 55, 58], or even the combination of the three strategies including contextual modelling [59]. However, these combinations may be risky since a pre-filtering strategy can cause loss of valuable information thus affecting accuracy [4].

Heuristic-based approaches are almost evenly distributed between the application of pre-filtering and contextual modeling strategies to realize context-aware recommendations. Regarding the application of pre-filtering, data sources are usually partitioned by context factors to improve data uniformity, which leads to stronger user/item similarities, as well as better confidence and support measures for association rules, thus improving the relevance of recommendations. In the case of contextual modeling, context information modifies how similarity is calculated.

With respect to contextual information, we found that most of the studied collaborative filtering systems have time, social, and location as the predominant factors. Furthermore, the application domains to which the surveyed systems are commonly applied are movies, restaurants, music, points of interests, social networks and e-retailing.

5.3. Hybrid approaches

Since hybrid approaches combine collaborative filtering and content-based recommendation methods in many different ways, there is not a unique abstract process that can characterize hybrid solutions the way we previously did for the non-hybrid processes depicted in Figs. 1 and 2. Table 4 presents the characterization of hybrid approaches, emphasizing on the way context is exploited.

As we found only five papers documenting hybrid RS, it is impossible to generalize their findings. Each approach follows its own strategy.

Table 4: Characterization of hybrid approaches

Appr.	Techniques	Paradigm	Context types	Domains	Means to incorporate context
[107]	**Content-based Profile representation** Item features	Pre-filt.	Time, Location	Movies, Music	Associate ratings with content-based attributes used to describe both user preferences and item features, and with the contextual factors gathered from the user experience (e.g., time of the day). Over the resulting vector space, the authors propose the application of several types of machine learning classification models.
	Collaborative filtering Model-based, **Tech.:** Naïve Bayes, Random forest, Multilayer Perceptron, and Support Vector Machine	Cont. Model.			
[108]	**Collaborative filtering User sim:** K-means	Pre-filt.	Location	Music, Points of interest	Takes into account user demographics: the geographical distance between the user and the event, and the subsequent time that it would take the user to arrive. It segments users into clusters, with every user having a probability of belonging to every cluster, and with each cluster having a probability distribution of liking every item. A discriminant filter evaluates the utility of the item for the user, considering a particular context.
	Content-based Profile representation Item features, **Discr. filter** Heuristic	Cont. Model.			
[28]	**Collaborative filtering User sim:**Pearson correlation, **Heuristic-based** Sum of products	Pre-filt.	Time, Location, Activity, Artificial (environ- ment)	Movies, Music, News	Performs contextual recommendations by combining a discriminant filter with an a aggregation of the ratings of similar users. A similarity measure between users takes into account their contextual profile.
	Content-based Profile representation Item features, **Discr. filter** Cosine similarity	Pre-filt.			
[109]	**Content-based Profile representation:** Item features	Cont. Model.	Social	Web Services	Identifies a couple of reading "experts" whose opinions can be regarded as guidance for news recommendation to particular individuals. Further, integrates this "expert" model with the content information and collaborative filtering, and propose a hybrid recommendation framework.
	Collaborative filtering Model-based, **Tech:** Matrix Factorization	Cont. Model.			
[110]	**Collaborative filtering Model-based**	Cont. Model.	Social, Location, Time	Social Networks	Social context is taken into account by considering the groups to which users belong to on an events-based social network. Users and events are described by the hour at which users attend events (time), and are compared by applying cosine similarity. Geographical preference of events is modeled by obtaining a probability density per user, taking into account the densities of attended events.
	Content-based Profile representation Item features, **Profile comput.** TF-IDF **Discr. filter** Cosine similarity	Pre-filt.			

ACCEPTED MANUSCRIPT

5.4. Findings in the exploitation of context information

Figure 3 summarizes general findings related to the exploitation of context by the systems described in the surveyed articles.

Figure 3: Summary of findings in the exploitation of context information

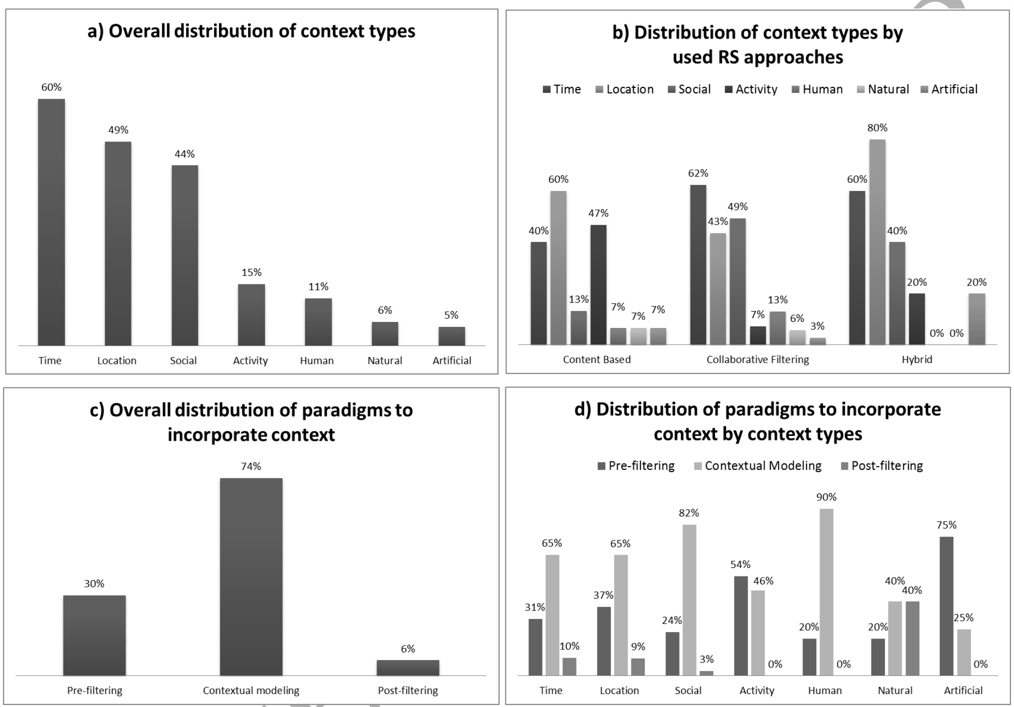


Figure 3.a presents the overall distribution of context types. According to this chart, time is the most used context factor followed by location and social information, whereas artificial is the less exploited context type followed by natural, human and activity. In the studied approaches, artificial context refers to data gathered from mobile sensors, natural context refers to weather conditions, and human context corresponds to user age, gender, mood, intent of purchase, preferences and hunger level. Only papers exploiting social context comment on the reasons why the exploited context type was selected. We hypothesize that, besides being relevant in all application domains, the main reason why time is the most exploited context type is that it is the easiest one to acquire: every system records information about transaction

dates, without requiring the explicit approval of users. As time context, location is also highly relevant and easy to acquire, however, its acquisition and usage, as in the case of social, activity and human context, requires user explicit approval. Artificial context does not necessarily compromise user privacy, however, its acquisition requires physical sensing infrastructures that are not always available.

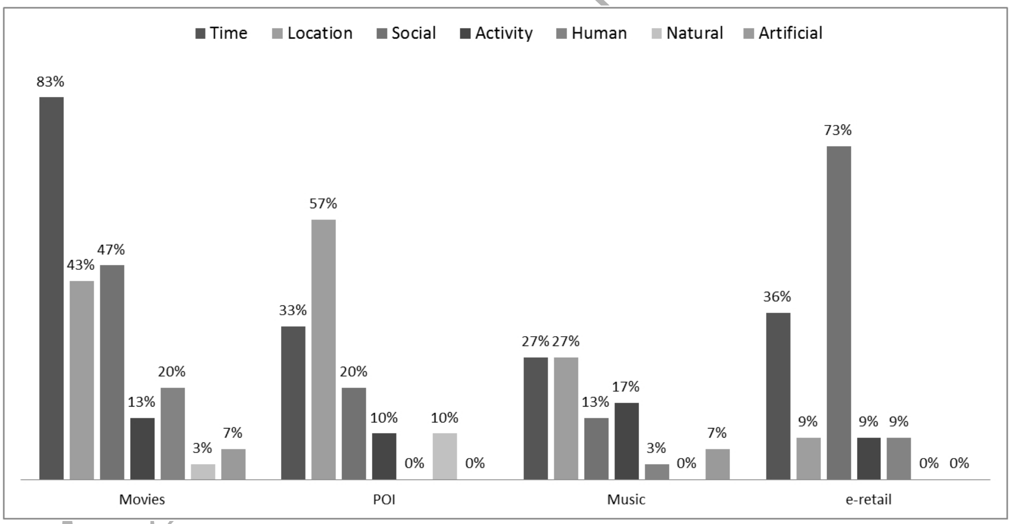
Regarding the context types used with the different recommendation approaches (i.e., content-based, collaborative filtering and hybrid), it is important to highlight that (cf. Fig. 3.b): i) only 13% of the content-based RS exploit social context. This is expected since social context emerges from the relationships among users, which are less relevant in content-based approaches; ii) location and activity are the most used context types in content-based RS. A reason for this is that the relationships existing between users and items usually emerge from the place where the item is used or bought, and the activity the user is performing while using an item. In addition, items are easily associated with places and activities; iii) time is the most exploited context type in collaborative filtering systems. This is probably associated with its easy acquisition, which becomes more relevant in collaborative filtering where it is required to characterize users under similar context situations; and iv) as expected, human context is more relevant in collaborative filtering than in content-based approaches, probably because demographic information is highly used in the analysis of user similarities.

Without doubt, contextual modeling, recognized by its effectiveness in improving the performance of recommendations, is the most common paradigm used to incorporate context into RS (cf. Fig. 3.c). Post-filtering, as discussed in previous subsections, is the less used, since its application may result on the discarding of time and space wise costly recommendations. Concerning the distribution of paradigms to incorporate context by context types (cf. Fig. 3.d), it is worth pointing out that systems exploiting activity (13 papers) and artificial (4 papers) context have pre-filtering as the predominant paradigm to incorporate context.

Most popular application domains identified in the studied papers are movies (30 papers, 34%), points of interest (POI, 18 papers, 21%), music (15 papers, 17%), and e-retailing (11 papers, 13%). Other domains are hotels & tourism (6 papers, 7%), web services (5 papers, 6%), news (4 papers, 5%), food (3 papers, 3%), indoor shopping (2 papers, 2%), social networks (2 papers, 2%), and e-learning (1 paper, 1%). Seven of the studied papers do not report targeting particular application domains (general application).

Figure 4 presents the distribution of context types by application domains. Movies is the only domain that exploits all context types, being time, social, and location the most exploited ones. As expected, location is the most common context type in the points of interest domain, followed by time. Concerning the music domain, location, time and activity are the most used context types. Activity is more predominant in this domain than in the others, probably because music genres are commonly associated with specific user activities. In the e-retailing domain, social is the predominant context type, followed by time. Here it is evident the influence of collaborative filtering as the predominant type of recommendation algorithm, particularly in this domain. Context types location, activity and human are equally exploited in e-retailing applications. Finally, it is worth also noticing that natural context, which in general refers to weather conditions, is more used in points of interest applications.

Figure 4: Distribution of context types by most popular application domains



6. Characterization of validation methods

The improvement of user experience is the ultimate goal of a recommender system. In order to measure it, a series of properties, each with a set of metrics, have been proposed and used since the first developments in the field. These properties allow us to determine the pertinence of the recommendations being suggested. Instances of these properties are predictive power, confidence, diversity, learning rate, coverage, scalability and user evaluation [111].

In this section we summarize the properties that were considered to evaluate the recommendation systems documented in the surveyed papers, particularly predictive power, which is the most commonly used evaluation property. The first two parts of this section focus on prediction metrics and evaluation protocols identified in the studied articles. Then, we summarize other properties that were also used to assess the quality of recommendations in the studied CARS. Finally, we present the list of datasets that we identified in our survey.

6.1. Prediction metrics

Among the different metrics that can be considered to evaluate RS, the most commonly used is predictive power. This could relate to the information retrieval origins of RS. All but five of the papers we surveyed use some kind of prediction metric to assess the quality of their recommendations.

Table 5 presents the distribution of the reviewed articles with respect to prediction metrics. The first column represents the class of metric. The second column refers to the specific prediction metric techniques, grouped by their class. The third column presents the number of papers that use the metric to validate the proposal, which are listed in the last column. It is important to note that some articles may use more than one prediction metric to evaluate their approach. We borrowed the definitions of these metrics from [111] and [112].

Prediction metrics are based on different types of comparisons between the recommended items and the accessed or consumed items. As mentioned in [111], there are three classes of prediction metrics: rating prediction, usage prediction and ranking metrics (cf. first column of Table 5).

Table 5: Metrics used to evaluate predictive power

Class	Prediction Metrics	#Approaches	Approaches' references
Rating prediction metrics	MAE	27	[4, 9, 42, 46, 47, 50, 51, 59, 63, 69, 73, 77, 79, 80, 84, 88, 89, 91, 92, 93, 96, 100, 101, 102, 103, 105, 106]
	RMSE	24	[9, 10, 47, 56, 71, 77, 78, 79, 80, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 100, 101, 105, 106]
Usage prediction metrics	Precision	43	[4, 10, 24, 27, 28, 29, 32, 33, 36, 39, 40, 41, 42, 43, 44, 47, 51, 53, 54, 56, 58, 60, 62, 63, 65, 66, 67, 70, 71, 72, 75, 80, 82, 83, 84, 95, 97, 98, 101, 103, 106, 107, 109]
	Recall	28	[4, 10, 29, 33, 40, 42, 43, 44, 50, 53, 54, 58, 62, 63, 65, 67, 70, 71, 72, 75, 82, 84, 97, 98, 101, 103, 106, 109]
	F-measure	10	[4, 10, 29, 33, 40, 43, 57, 62, 67, 97]
	AUC	5	[24, 60, 74, 103, 107]
	MAP	8	[24, 34, 55, 76, 79, 95, 98, 103]
	BR	1	[95]
Ranking metrics	NDCG or DCG	9	[30, 31, 35, 47, 57, 72, 74, 87, 110]
	Hit Ratio	6	[48, 68, 70, 74, 87, 99]
	MRR or CRR	3	[99, 103, 104]
	Map@K	1	[101]
	Rt10	1	[35]
	None or other type reported	5	[37, 45, 49, 52, 81]

Rating prediction metrics. These metrics measure the correctness of the recommendations in terms of their error. The two metrics we identified in the studied articles are *root mean squared error* (RMSE) and *mean absolute error* (MAE). These metrics measure the distance between predicted and real ratings. So, lower values of RMSE and MAE indicate a higher predictive power. Since RMSE squares the error, it tends to penalize large errors more heavily. The choice between RMSE and MAE is at discretion of the developer. For instance, in the movies domain, while in [85] the RS is evaluated by measuring the quality of suggestions using RMSE, giving more importance to larger differences between the predicted and real ratings, in [73] the evaluation is based on MAE, considering a linear approach to measure the errors.

Usage prediction metrics. These metrics are based on different types of proportions between recommended and consumed items, as determined by the contingency table that compares them. The following are the usage prediction metrics that we identified in the surveyed papers:

- *Precision (or true positive rate)* measures the proportion of recommended items that result relevant to the users, that is, those recommended items that the user actually consumes. The CARS proposed in [39] is evaluated with respect to a context-free approach using this metric. This system exploits user location (i.e., a gym, the library, the office, the transportation system) to suggest appropriate songs. The results show that the proposed approach outperforms its baseline (e.g., a precision of 60% and 50%, respectively, in situations where the location context corresponds to the transportation system).

- *Recall (or sensitivity)* measures the proportion of consumed items that were correctly recommended, that is, the fraction of items relevant to the user that were suggested by the system. Recall and precision are usually considered together as two facets of the quality of the recommendation. An example is presented in [53], where precision and recall are used as the basis to show that the greater the cardinality of a set of recommended items is, the higher the value of recall is.

- *Specificity (or true negative rate)* measures the proportion of not recommended items that are irrelevant to the users. This metric was not directly used in any of the surveyed papers, but it is a basis for the definition of other metrics such as AUC, explained below.

- The *F-measure* family of metrics combines precision and recall, allowing for the comparison of different RS using a single metric. Adomavicius et al. [4] use this metric to compare the effects of taking into account independent context factors (i.e., social, time and location), or combinations of them, when predicting user ratings. The results showed that the segments theater-weekend (i.e., location-time), theater (i.e., location), and theater-friends (i.e., location-social) substantially outperform the standard methods in terms of F-measure. They also applied F-measure to show how their approach outperforms regular non-context RS.

- *AUC (or area under the curve*) is a more robust metric that considers the variations between the true positive rate (recall) and the true negative rate (1 - specificity). The movie CARS published in [74] is evaluated using this metric.

- Other usage prediction metrics are refinements of simpler ones, such as *mean average precision* (MAP) [55], or *benefit ratio* (BR) [95]. The latter is defined as the ratio between the number of users who get an improved prediction and the number of users who get a deteriorated prediction.

Ranking metrics. These metrics assume that the utility of a recommended item is proportional to its position in the ordered list of recommendations produced by the RS. The ranking metrics used to evaluate the CARS included in our survey are the following:

- *Normalized discounted cumulative gain (NDCG) and discounted cumulative gain (DCG)* consider that highly ranked relevant objects give more satisfaction than poorly ranked ones. Biancalana et al. [30] use NDCG to compare their CARS performance with the performance of other approaches. They also study the effect on the quality of recommendations, as measured by NDCG, by taking into account different contextual factors separately. Biancalana et al. [30] and Hong et al. [53] argue that CARS produce better results when the number of items to recommend increases.

- *Hit ratio* measures whether a user's target choice appears in the top-K recommendation list. Generally denoted as Hit@K, where K indicates the number of recommended items. Unger et al. [87] find that the use of latent context models provides a noticeable advantage over non-contextual models for almost every value of K. The advantage is greater with small values of K (i.e., ranging from 1 to 4), which means that the latent context model is highly capable of ranking a suggested recommendation according to the user's current context.

- *Mean reciprocal rank (MRR) and Cumulative reciprocal rank (CRR)* evaluate the ranking position of a user's target choice in the recommendation list. Chen and Chen [99] use CRR to evaluate recommendations that take into account location context.

- *Mean average precision (MAP@K)* considers the precision of the first K recommended ranked items. Every item on the list of ranked items contributes to the MAP@K measure of the recommendation proportionally to its position, if they were indeed accessed/consumed by the user for which the recommendations were made. Jiang et al. [101] use this metric, along with other metrics (MAE, RMSE, Precision, Recall, F1 measure) to evaluate the performance of different configurations of their proposed model.

- *Rt10* averages the ratings of the top 10 recommended items. It is used specially in information retrieval. Son et al. [35] show, using the Rt10 metric, that news article recommendations are more effective when considering their particular geographical location.

Finally, from the five papers that do not report the usage of a particular prediction metric, two of them use other mechanisms to evaluate their models. For instance, to evaluate user satisfaction, Hong et al. [37] measure effectiveness and usability, whereas Baltrunas et al. [45] use a standard usability questionnaire. The approach presented in [81] is compared to a baseline model in terms of accuracy without reporting any metrics. However, these authors published the same model in [82] including a quantitative evaluation.

6.2. Evaluation protocols

This subsection presents the different evaluation protocols applied by the authors of the surveyed papers. These protocols define the way data sets are handled and partitioned into training and test sets to evaluate the quality of the recommendations. We found that in all reported cases context was consistently considered as a data set partitioning criterion, and that the baseline approach is usually a context-free RS, or a CARS that follows a different approach than the one being proposed.

Table 6 presents the distribution of reviewed articles with respect to evaluation protocols. The first column lists the evaluation protocols, the second column shows the number of papers that use the protocol to validate the proposed CARS, and the third column specifies each of the corresponding surveyed papers. Papers [33, 41, 56, 103] did not report on the used evaluation protocol.

Table 6: Evaluation Protocols

Evaluation Protocols	#Approaches	Approaches
Holdout or cross-validation	46	[9, 10, 28, 31, 32, 33, 39, 45, 48, 54, 57, 58, 59, 60, 62, 63, 65, 66, 67, 68, 69, 70, 71, 72, 75, 76, 78, 79, 81, 82, 84, 87, 91, 92, 93, 94, 96, 97, 100, 101, 102, 104, 105, 106, 109, 110]
K-fold cross validation	21	[4, 30, 34, 38, 42, 43, 44, 47, 51, 55, 73, 77, 80, 83, 86, 88, 89, 90, 95, 98, 107]
Hypothesis test	5	[27, 35, 40, 46, 99]
Bootstrapping	2	[4, 29]
Simulation	1	[64]
None reported	4	[33, 41, 56, 103]

Holdout or cross-validation. This is one of the most commonly used evaluation protocols. It consists in splitting the dataset into two sets: training (e.g. 70% of the data) and test (30%). The recommendation model/algorithm is trained using the first set, and evaluated using the second one. The training and test data can be obtained in different ways, depending on the application domain and the way context information affects the recommendations. For example, in [39], Cheng and Shen evaluate their music CARS by splitting the data set according to time and location context, before extracting the training and test sets.

K-fold cross-validation. This is a more sophisticated evaluation protocol that consists in partitioning the dataset into a K equally sized groups of items called folds, to then perform a cross-validation evaluation process. One of the folds is chosen as the test set and the union of the other folds as the training set. This process is repeated K times, each time changing the fold used as test set. This evaluation protocol is used to evaluate the CARS presented in [42]: for each fold, the authors compute the MAE, precision and recall metrics, and average their results to then estimate the quality of their recommendation model. The CARS proposed in [86] and [4] apply independent recommendation processes for each relevant context. The authors evaluate the performance of these systems using K-fold cross-validation. This allows them to compare the predicted ratings for each context, and establish the contexts for which the recommendation is more accurate.

Hypothesis test. This protocol uses statistical inference. It is based on the computation of the statistical significance of the differences between the compared CARS. In particular, it is useful to identify whether there is a significant difference between contextual and non-contextual recommendations. The CARS presented in [99] is evaluated using this protocol, where the hypothesis is that user preferences are influenced by contextual factors, and that the proposed recommendation algorithm is capable of capturing such influences. For example, user restaurant preferences may not be influenced solely by aspects such as food quality, value, and service, but also by contextual factors such as location.

Bootstrapping. This protocol relies on random sampling with replacement. That is, a subset of size N is taken from the original data set and then partitioned into training and test data. This process is repeated multiple times, considering always the whole original data set as the basis for the re-sampling. The estimation of the performances of the RS is finally aggregated from the results of each re-sample. For instance, Musto et al. [29] use a bootstrapping-based protocol proposed in [4]. This protocol consists in identifying different possibly overlapping subsets of the dataset based on context types (e.g., establishing a contextual segment composed of time context observations, or another one composed of location context observations). The authors extract 500 random re-samples from their dataset and split them by assigning 29/30th of the items to the training set and 1/30th to the test set. They use precision, recall and F1 as the metrics to evaluate the performance of their system with respect to the different contextual segments.

Simulation. When there is no dataset available upon which to perform the evaluation of the recommendation model, it is possible to generate an artificial synthetic dataset using simulation techniques, based on certain suppositions (e.g. normal distributions). Eirinaki et al [64] applied this method to generate a social network simulating trust relationships between users (social context), and the matrix relating users to items (in their case, web services).

6.3. Other properties

Predictive metrics measure how close predicted preferences are from user real preferences. However, predictive power is not enough to measure whether the recommendation was satisfactory, useful or effective to the users [112]. A recommendation system may be highly accurate, but only for those items for which a recommendation may result useless (e.g., products that the user buys very frequently).

Table 7 presents the approaches that consider properties other than predictive power to evaluate the proposed CARS. The plus sign in a cell indicates that the corresponding property is used to evaluate the CARS proposed in the paper represented by the row (cf. first column of the table). As in the case of the prediction metrics presented above, we borrowed the definition of these properties from [111] and [112].

Table 7: Other properties

Appr.	Learning rate	Confidence	Diversity	Novelty	Coverage	Scalability	Usability
[76]	+						
[98]	+						
[86]	+						
[90]	+	+	+				
[62]		+				+	
[94]			+		+		
[66]				+			
[53]						+	
[79]	+					+	
[99]	+						+
[30]							+
[27]							+
[38]							+

Learning rate. This property measures how fast an algorithm produces good recommendations. Learning rate is also associated with the parameter that determines how fast or slow a recommendation model will converge towards an optimal solution. We found that all of the CARS evaluated through this property are based on model-based strategies (i.e., matrix and tensor factorization, and linear regression), and exploit context information by implementing the contextual modeling paradigm.

Confidence. This property refers to the trustworthiness of the system predictions, and the extend to which they help users make more effective decisions. The work published in [90] uses this property to evaluate, under specific contexts, the quality of several prediction models based on matrix factorization,

Diversity. This property measures how dissimilar are the recommended items among them. It is defined as the opposite of similarity. Zhang et al.

evaluate the quality of their movie CARS in terms of diversity [90]. They argue that a good recommender system is the one that delivers considerable different recommendations, for example, films belonging to different genres.

Novelty. Based on the assertion that the relevance of a recommended item depends not only on its correctness, but also on its novelty. Nocera et al. [66] define an ad-hoc measure that takes into account whether the recommended items were already known to the user (e.g. accesed in the past).

Coverage. This property measures the proportion of items that the system recommends from the universe of available items. Not all of the available items are subject to be recommended. This is the case of collaborative-filtering RS for items that have not been yet consumed or rated by the users. Sitkrongwong et al. measure accuracy and coverage for different contextual factors [94]. They found that, since not every context applies to all items, it is possible to increase the coverage by ignoring some of the relevant contextual factors. Nevertheless, there is a trade-off between accuracy and coverage that can be mitigated by identifying the set of relevant contextual factors for each user and each item separately, instead of identifying the relevant contextual factors for the entire data set.

Scalability. This property refers to the computational capability of the recommender system to handle a growing amount of data. Khalid et al. address this property by storing and processing data on geographically distributed nodes [62]. Shi et al. measure scalability in terms of time complexity [79]. We did not find any relation between context and scalability.

Usability. This property measures the satisfaction of the user with respect to the ease of use of the RS. In [27], Hawalah and Fasli evaluate usability through a questionnaire that asks users to rate a set of statements, including some to evaluate the contextual nature of the system: i) *the items recommended to me matched my interests*, ii) *the items recommended to me took my personal context requirements into consideration*, and iii) *I was only provided with general recommendations.*

6.4. Data sets

Table 8 characterizes the 16 data sets that we identified as publicly available from 32 out of the 87 characterized papers. For each data set, we indicate the papers that use it, the domain, and the supported context types.

Table 8: Data sets identified in the SLR

Appr.	Domain	Brief description	Context types	URL
[73]	Movies	Information about movies, users and ratings.	Human (age, gender)	https://research.yahoo.com
[9, 50, 57, 60, 75, 78, 80, 90, 97]	Movies	MovieLens: information about ratings, users, and items (movies).	Human (age, gender, occupation), Time (day, month, year, hour, minute, second)	http://grouplens.org/datasets/movielens
[72, 86, 88, 94]	Movies	Data set collected for experiments using an on-line application for rating movies. Users fill in a simple questionnaire created to explicitly acquire the contextual information describing the situation during the consumption. It contains records of users, ratings and movies.	Time (season, day type), Location, Natural (weather), Social	http://students.depaul.edu/yzheng8/DataSets.html
[85]	Movies	Provided by the Netflix Prize. It contains records of ratings, users, and movies.	Time	http://www.netflixprize.com
[24]	Movies	CAMRa 2011s MoviePilot Dataset: contains ratings, users, and items.	Time	http://2011.recsyschallenge.com/dataset
[36, 48]	Music	Information about users, artists, bi-directional user-friend relations, and user-listened artist relations	Social, Time (day, month, year)	http://grouplens.org/datasets/hetrec-2011
[54, 58, 62, 65, 100, 106]	Points of interest, Hotels & Tourism	Data set acquired from FourSquare. It contains information places.	Location, Social	https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[68]	General application	Information about users, tagged papers, and tags.	Time, Social	http://www.citeulike.org/faq/data.adp
[69]	Movies	Provided by the Comaq Systems Research Center. Ratings given by users to movies.	Time	http://www.research.compaq.com/SRC/eachmovie
[65]	Points of interest, Hotels & Tourism	Friendship network with information about locations and user check-ins (user, check-in time, latitude, longitude, location)	Social, Location	http://snap.stanford.edu/data/loc-gowalla.html
[57]	General application	Information of ratings given by users to jokes	Human (user preferences)	http://eigentaste.berkeley.edu/dataset/
[71, 91, 93, 105]	E-retailing	Information about reviews of products done by users	Social	http://www.trustlet.org/epinions.html
[70, 93]	E-retailing, Music	Information about reviews of products done by users	Social, Time	https://labrosa.ee.columbia.edu/millionsong/lastfm
[91, 92, 93, 105]	E-retailing, Books, Music, Movies	Information about user reviews and recommendation services for movies, books, and music	Social	http://socialcomputing.asu.edu/datasets/Douban

7. The effect of incorporating context into RS

When conducting an SRL on CARS, a natural question is the level of improvement of RS performance (e.g., in terms of accuracy) obtained from the inclusion of a particular context type into the recommendation process. Nevertheless, answering this question results impractical, given the wide spectrum of recommendation techniques that can be combined with the different context types, through any of the three existing paradigms to include context information into RS. Furthermore, the performance of these systems vary depending on the used dataset and evaluation metrics, which make the results incomparable. For this reason, questions such as *what is the context type that provides the best results for improving recommendations in a particular context domain?* were not included in the set of research questions that drove the development of this SLR.

Despite the limitations to compare the effectiveness of particular context types, we surveyed the impact of incorporating context information into the reported systems. We found that only 36 out of the 87 studied articles quantitatively evaluate the obtained improvements with respect to baseline approaches (cf. Table 9). This constitutes an opportunity for this research community—formal validations and benchmarks of CARS are of paramount importance to advance this field. The systems reported in these 36 papers were all evaluated with respect to at least one baseline approach in terms of accuracy, through any of the metrics listed in Table 5.

Table 9 presents the improvements reported by these papers. For each approach (cf. Column *Appr.*) the table includes the types of context exploited by the corresponding CARS, the application domain, and the improvement obtained for each of the used metrics. The table groups accuracy metrics according to the three metric categories (i.e., usage prediction, rating prediction and ranking prediction), explained in Sect. 6.1. The goal of this table is to report the surveyed information rather than to provide a basis for comparing the improvements obtained in RS when including the different context types.

Table 9: The effect of incorporating context into the RS that were evaluated quantitatively

Appr.	Types of context	Application domains	Usage Prediction				Rating Prediction		Ranking Prediction
			Precision	Recall	F-Measure	MAP	MAE	RMSE	NDCG, Hit Ratio, MRR
[67]	Human(mood), Time	e-learning	2%	2%	5%				
[50]	Time, Location	Movies		22%			32%		

Table 9: The effect of incorporating context into the RS that were evaluated quantitatively

Appr.	Types of context	Application domains	Usage Prediction				Rating Prediction		Ranking Prediction
			Precision	Recall	F-Measure	MAP	MAE	RMSE	NDCG, Hit Ratio, MRR
[90]	Human (age, gender)	Movies						3%	
[99]	Social, Time	Hotels and Tourism							
[79]	Human (hunger level), Time, Location	Food				15%	9%	9%	
[80]	Social, Time	E-retailing, Movies	6%				17%	14%	
[47]	Location, Time	Point of interest	Between 1,7% and 3,1%				9%	4%	
[51]	Time, Location, Social	Movies	10%						
[72]	Time, Location, Natural (weather), Social	Movies, Hotels and Tourism	Between 80% and 200%; and between 16% and 103%						
[29]	Time, Social, Location	Movies			About 10%				
[98]	Time, Location, Social	Movie	Between 2% and 42%			Between 2% and 6%			
[43]	Time, Location	Music, Point of interest	Between 5% and 33%	Between 5% and 33%	Between 5% and 33%				
[75]	Time	Movies		Between 30% and 35%					
[73]	Human(age, gender), Time, Social	Movies					Between 5% and 30%		
[84]	Social	Not Identified	Between 12% and 22%		About 21%			About 24%	
[76]	Location, Activity	e-retailing	About 53%			About 40%			
[87]	Location, Time, Activity	Point of interest							Hit ratio: About 25%
[68]	Time, Social	General application							Hit ratio: Between 34.56% and 35.91%
[93]	Social	E-commerce					About 10%	About 10%	

Table 9: The effect of incorporating context into the RS that were evaluated quantitatively

Appr.	Types of context	Application domains	Usage Prediction				Rating Prediction		Ranking Prediction
			Precision	Recall	F-Measure	MAP	MAE	RMSE	NDCG, Hit Ratio, MRR
[91]	Social	Books, Music, Movies					Between 9% and 18%	Between 7% and 17%	
[58]	Social	Books, Music, Movies	Avg: 73.27 times better	Avg: 73.27 times better					
[69]	Time	Movies					About 5%		
[92]	Social	General application					Avg: 21%	Avg: 18%	
[31]	Human (user interest)	Web services							NDCG: 40%
[32]	Social	Multimedia	About 25%						
[100]	Social, Location	Points of interest, Hotels & Tourism					Best case: 22%	Best case: 35%	
[65]	Social, Location	Points of interest, Hotels & Tourism	Best case: 15%	Best case: 10%					
[110]	Social, Location, Time	Social networks & Tourism							NDCG: 60%
[101]	Social	General application					Between 10% and 27%		
[59]	Location	Web services					Between 2% and 3%		
[102]	Time	Web services					Between 5% and 20%		
[104]	Social	E-retailing							MRR: between 8% and 25%
[56]	Social	E-retailing	Best case: 78%						
[57]	Different types of context	General application	Best case: 78%						DCG: Between 2.5% and 5%
[106]	Social, Location	Points of interest					Best case: 12.6%	Best case: 14.5%	
[105]	Social	E-retailing					Best case: 16.24%	Best case: 16.09%	

8. Research opportunities

This section provides CARS researchers with a list of research opportunities, most of them borrowed from the studied articles. From each paper, we identified, categorized, and analyzed the challenges that authors defined as worthy of future work. Each subsection corresponds to one of the nine challenge categories that we identified: *dynamic context management, context gathering, context reasoning, contextual modeling, problems inherent in RS, CARS evaluation, users in the loop, self-adaptation* and *privacy and ethical considerations.*

8.1. Dynamic Context Management

Traditional CARS assume that context information is immutable over time, even when user situations continuously change. Evidence of this are deal recommendation systems that keep sending offers to the user for events currently happening in her home city, despite she is in a several day business trip that is scheduled in her agenda, and the user's agenda as well as her current location can be easily monitored by modern applications [7]. This static vision of context information causes that RS deliver recommendations that are irrelevant to users, which has negative effects for businesses.

To deal with this dynamic nature of context, CARS must be equipped with runtime mechanisms to identify relevant context and integrate it into the recommendation process dynamically [3, 5]. This implies also to enable RS to manage the life cycle of context information at runtime, for instance, to identify context variables that become relevant or irrelevant, and treat them accordingly. For example, by adapting the recommendation model according to new context variables that may become relevant while the user interacts with the system.

Dynamic context management research in RS includes investigating mechanisms to i) identify context changes that affect the relevance of recommendations; ii) characterize the life cycle and dynamics of context information; and iii) develop situation-aware and self-adaptation mechanisms to enable CARS with the ability to adjust recommendation models at runtime. Among the studied papers, [3, 28, 30, 36, 47, 69, 85] declare dynamic context and its management challenges as a future research area.

The following two categories of research opportunities, context gathering and context reasoning, are completely related to dynamic context management, since they are concrete phases of the context information life cycle [5].

8.2. Context Gathering

Context gathering refers to the process of acquiring context information from the user's environment. When the relevant context is dynamic (e.g., context that changes over time such as the purchase intent of a user), context acquisition requires automatic mechanisms to detect context sources that become available at runtime, and deploy the sensors required to gather this information. Context gathering challenges include: i) the acquisition of context information from non-explicit and non-traditional context sources (e.g., to identify user intents and motivations); and ii) the development of user interfaces that allow the acquisition of relevant context, without requiring user explicit inputing through traditional interfaces. The authors of the following papers highlight the importance of context gathering research [27, 40, 45, 48, 60, 76, 84, 96].

8.3. Context Reasoning

Context reasoning refers to the inference of implicit context facts from raw context [5]. When context is highly dynamic, context management mechanisms must support the addition of reasoning rules dynamically. Context reasoning challenges in RS include: i) inferring context facts from the combination of different context variables; ii) understanding, particularly at runtime, the relationships between context situations and user preferences; and iii) exploiting context available in user profiles effectively. Authors of papers [4, 30, 39, 45, 52, 58, 82, 86, 107, 108] identify context reasoning as a relevant research topic.

8.4. Contextual Modeling

Pre-filtering, contextual modeling and post-filtering are the three existing paradigms to incorporate context into RS. In contextual modeling, context information is directly integrated into the recommendation model, which, in many cases, has been proved to be more effective than pre- and post-filtering approaches. As a result, an important number of researchers investigate how to exploit context information through contextual modeling [4, 24, 30, 41, 43, 51, 56, 68, 73, 79, 81, 86, 91, 105, 106]. Contextual modeling challenges include the development of new techniques and mechanisms to: i) integrate context into traditional recommendation models; ii) improve rating estimation methods by exploiting context; and iii) identify the context variables that must be integrated into the recommendation model.

8.5. Problems inherent in RS

Context information can be also useful to solve specific problems in RS. Such is the case of the cold-start, self-biased recommendations, and sparsity problems. Concerning the cold-start problem, context provides information that allows the characterization of users, even when they are newcomers to the system [38, 39, 93, 109]. Regarding the self-biased problem, an important challenges is to develop mechanisms to prevent the self-influence of frequently recommended items on future recommendations; the approach presented by Nocera et al. [66] deals with this problem using a novelty metric that considers social context. Concerning the sparsity problem, context-dependent matrices could help decrease sparsity by taking into account different subsets of dimensions under particular context situations [56, 59, 88, 92, 93, 101, 102, 109]. For example, to infer user ratings in a department store, instead of taking into account all of the products the user has bought in the past, one could use only those products directly associated with the user's current purchase intent (e.g., vacation planning, back to school season).

8.6. CARS evaluation

The evaluation of new methods and techniques is crucial to advance the state of the art of CARS, and to confidently apply new developments in real life. Major evaluation challenges identified from the studied papers are [29, 36, 46, 51, 64, 69, 76, 107]: i) the investigation of new properties and metrics; ii) the development of benchmarks that facilitate the understanding of approaches that perform better in particular circumstances; iii) the development and documentation of real life experiments in different application domains; and iv) the acquisition of contextual real data to improve the quality of validations.

8.7. Users in the loop

There is an increasing tendency to conceive users as part of software systems, instead of entities that simply interact with systems. This is commonly known as *the integration of users in the loop*. Users can be integrated in the recommendation process, at one or several of its phases, for example, through feedback that can be used to improve recommendations. Users in the loop are also valuable sources of relevant context. An important challenge is to achieve a seamless integration to avoid affecting the natural behavior of the user. This challenge category was explicitly addressed in [50].

8.8. Self-adaptation

Self-adaptive software systems adjust their structure or behavior at runtime to control the satisfaction of functional and non-functional requirements [113]. To achieve these dynamic capabilities, these systems are instrumented with feedback loops that measure outputs and compare them against reference inputs. If the measure output does not correspond with the desired value specified in the reference input, a controller adjusts the target system to obtain better results [114]. An interesting research direction for the advancement of recommender systems is to instrument them with feedback loop-based mechanisms that allow them to self-improve at runtime. Authors of paper [33] highlight self-adaptation as a promising research direction. In particular, they are interested in implementing a feedback mechanism that adjusts the semantic similarity metric at runtime with the goal of improving performance.

8.9. Privacy and ethical considerations

Privacy and ethics are important aspects to be considered in CARS. Several relevant challenges arise from the need to assure these aspects, which is particularly difficult at runtime. For example, whenever a new context source is identified as relevant, how to validate with the user that this information can be used by the system, that this usage is transparent to the user, and that this information will be used only for the purposes approved by the user. Privacy and ethical aspects are of paramount importance to develop confidence and trust in the use of personalization in CARS [27].

9. Conclusions

This paper presented a comprehensive characterization of context-aware recommendation processes and systems, based on the findings of a systematic literature review (SLR) we conducted to survey CARS that were published between 2004 and 2016. This study was conducted with the goal of helping practitioners and researchers understand how context information can be effectively combined with recommendation mechanisms. The main results provide a clear understanding about where context information is usually integrated into the recommendation process, the techniques available to exploit context information depending on the underlying recommendation approach and the phase of the process where context is included, the context types more frequently exploited in the different application domains, and the most common used evaluation mechanisms, including properties, metrics and protocols.

Despite the comprehensiveness of this study, it is unfeasible to conclude about the effectiveness of using particular context types in specific application domains. This is in part because the effect of including context into RS is difficult to generalize given that the results depend on the nature of the used data sets and recommendation approaches. Furthermore, validation methods must be improved to include quantitative measures that allow a more objective evaluation of the proposed approaches—36 out of the 87 studied papers evaluate their systems quantitatively by comparing, against other approaches used as baselines, the improvements obtained with the integration of context information into the recommender system.

Besides the need for improving validation methods, this survey exposes also several research challenges that deserve further investigation. In particular, those related to the need for i) instrumenting CARS with runtime mechanisms to manage context dynamically along its life cycle; ii) developing new techniques to exploit context directly into the recommendation model; iii) exploiting context to solve inherent RS problems, in particular, the cold-start, self-biased recommendations, and sparsity problems; iv) instrumenting RS with self-adaptation capabilities, and v) solving user-oriented issues such as their better integration in the recommendation loop, as well as the privacy and ethical considerations that arise.

Acknowledgments

This work was funded by Universidad Icesi through its institutional research support program.

References

[1] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, Recommender systems handbook, Vol. 1, Springer, 2011.

[2] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, P. Steggles, Towards a better understanding of context and context-awareness, in: Handheld and ubiquitous computing, Springer, 1999, pp. 304–307.

[3] N. M. Villegas, Context management and self-adaptivity for situation-aware smart software systems, Ph.D. thesis, University of Victoria (2013).

[4] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, Incorporating contextual information in recommender systems using a multidimensional approach, ACM Transactions on Information Systems (TOIS) 23 (1) (2005) 103–145.

[5] N. M. Villegas, H. A. Müller, Managing dynamic context to optimize smart interactions and services, in: The smart internet, Springer, 2010, pp. 289–318.

[6] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 734–749.

[7] S. Ebrahimi, N. M. Villegas, H. A. Müller, A. Thomo, SmarterDeals: a context-aware deal recommendation system based on the SmarterContext engine, in: Proc. 2012 Conf. of the Center for Advanced Studies on Collaborative Research, IBM Corp., 2012, pp. 116–130.

[8] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: Recommender systems handbook, Springer, 2011, pp. 217–253.

[9] H. Ma, T. C. Zhou, M. R. Lyu, I. King, Improving recommender systems by incorporating social contextual information, ACM Transactions on Information Systems (TOIS) 29 (2) (2011) 9.

[10] U. Panniello, M. Gorgoglione, Incorporating context into recommender systems: an empirical comparison of context-based approaches, Electronic Commerce Research 12 (1) (2012) 1–30.

[11] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: A survey, Decision Support Systems 74 (2015) 12 – 32.

[12] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. rep., Keele University (2007).

[13] B. Sheth, P. Maes, Evolving agents for personalized information filtering, in: Proc. 9th Conf. on Artificial Intelligence for Applications, IEEE, 1993, pp. 345–352.

[14] K. Lang, Newsweeder: Learning to filter netnews, in: Proc. 12th Int. Conf. on machine learning, 1995, pp. 331–339.

[15] M. Pazzani, D. Billsus, Learning and revising user profiles: The identification of interesting web sites, Machine learning 27 (3) (1997) 313–331.

[16] W. Hill, L. Stead, M. Rosenstein, G. Furnas, Recommending and evaluating choices in a virtual community of use, in: Proc. SIGCHI Conf. on Human factors in computing systems, ACM Press/Addison-Wesley Publishing Co., 1995, pp. 194–201.

[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in: Proc. 1994 ACM Conf. on Computer supported cooperative work, ACM, 1994, pp. 175–186.

[18] R. Burke, Knowledge-based recommender systems, in: Encyclopedia of library and information systems, Marcel Dekker, 2000, p. 2000.

[19] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, E. Duval, Context-aware recommender systems for learning: a survey and future challenges, IEEE Transactions on Learning Technologies 5 (4) (2012) 318–335.

[20] A. Zimmermann, A. Lorenz, R. Oppermann, An operational definition of context, in: Modeling and using context, Springer, 2007, pp. 558–571.

[21] M. Kaminskas, F. Ricci, Contextual music information retrieval and recommendation: State of the art and challenges, Computer Science Review 6 (2) (2012) 89–119.

[22] Q. Liu, H. Ma, E. Chen, H. Xiong, A survey of context-aware mobile recommendations, International Journal of Information Technology & Decision Making 12 (01) (2013) 139–172.

[23] Z. D. Champiri, S. R. Shahamiri, S. S. B. Salim, A systematic review of scholar context-aware recommender systems, Expert Systems with Applications 42 (3) (2015) 1743–1758.

[24] P. G. Campos, F. Díez, A. Bellogín, Temporal rating habits: a valuable tool for rating discrimination, in: Proc. of the 2nd Challenge on Context-Aware Movie Recommendation, ACM, 2011, pp. 29–35.

[25] S. Inzunza, R. Juárez-Ramírez, A. Ramírez-Noriega, User and context information in context-aware recommender systems: A systematic literature review, in: New Advances in Information Systems and Technologies, Springer, 2016, pp. 649–658.

[26] S. Seifu, S. Mogalla, A comprehensive literature survey of context-aware recommender systems, International Journal of Advanced Research in Computer Science and Software Engineering 3 (4) (2016) 40–46.

[27] A. Hawalah, M. Fasli, Utilizing contextual ontological user profiles for personalized recommendations, Expert Systems with Applications 41 (10) (2014) 4777–4797.

[28] A. M. Otebolaku, M. T. Andrade, Context-aware media recommendations for smart devices, Journal of Ambient Intelligence and Humanized Computing 6 (1) (2015) 13–36.

[29] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, Contextual eVSM: A content-based context-aware recommendation framework based on distributional semantics, in: E-Commerce and Web Technologies, Springer, 2013, pp. 125–136.

[30] C. Biancalana, F. Gasparetti, A. Micarelli, G. Sansonetti, An approach to social recommendation for context-aware mobile services, ACM Transactions on Intelligent Systems and Technology (TIST) 4 (1) (2013) 10.

[31] B. Cao, J. Liu, M. Tang, Z. Zheng, G. Wang, Mashup service recommendation based on user interest and social network, in: 2013 IEEE 20th International Conference on Web Services, 2013, pp. 99–106.

[32] Z. Wang, L. Sun, W. Zhu, S. Yang, H. Li, D. Wu, Joint social and content recommendation for user-generated videos in online social network, IEEE Transactions on Multimedia 15 (3) (2013) 698–709.

[33] L. O. Colombo-Mendoza, R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández, J. J. Samper-Zapater, Recommetz: A context-aware knowledge-based mobile recommender system for movie show-times, Expert Systems with Applications 42 (3) (2015) 1202–1222.

[34] W. Lee, K. Lee, Making smartphone service recommendations by predicting users' intentions: A context-aware approach, Inf. Sci. 277 (2014) 21–35.

[35] J.-W. Son, A. Kim, S.-B. Park, et al., A location-based news article recommendation with explicit localized semantic analysis, in: Proc. 36th ACM SIGIR Int. Conf. on Research and development in information retrieval, ACM, 2013, pp. 293–302.

[36] D. Shin, J.-w. Lee, J. Yeon, S.-g. Lee, Context-aware recommendation by aggregating user context, in: Proc. 2009 IEEE Conference on Commerce and Enterprise Computing, IEEE, 2009, pp. 423–430.

[37] J. Hong, E.-H. Suh, J. Kim, S. Kim, Context-aware system for proactive personalized service based on context history, Expert Systems with Applications 36 (4) (2009) 7448–7457.

[38] X. Wang, D. Rosenblum, Y. Wang, Context-aware mobile music recommendation for daily activities, in: Proc. 20th ACM Int. Conf. on Multimedia, ACM, 2012, pp. 99–108.

[39] Z. Cheng, J. Shen, Just-for-me: An adaptive personalization system for location-aware social music recommendation, in: Proc. Int. Conf. on Multimedia Retrieval, ACM, 2014, p. 185.

[40] B. Fang, S. Liao, K. Xu, H. Cheng, C. Zhu, H. Chen, A novel mobile recommender system for indoor shopping, Expert Systems with Applications 39 (15) (2012) 11992–12000.

[41] M.-H. Kuo, L.-C. Chen, C.-W. Liang, Building and evaluating a location-based service recommendation system with a preference adjustment mechanism, Expert Systems with Applications 36 (2) (2009) 3543–3554.

[42] L. Baltrunas, F. Ricci, Experimental evaluation of context-dependent collaborative filtering using item splitting, User Modeling and User-Adapted Interaction 24 (1-2) (2014) 7–34.

[43] M. A. Domingues, A. M. Jorge, C. Soares, Dimensions as virtual items: Improving the predictive ability of top n recommender systems, Information Processing & Management 49 (3) (2013) 698–720.

[44] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, J. Tian, Mining mobile user preferences for personalized context-aware recommendation, ACM Trans. Intell. Syst. Technol. 5 (4) (2014) 58:1–58:27.

[45] L. Baltrunas, B. Ludwig, S. Peer, F. Ricci, Context relevance assessment and exploitation in mobile recommender systems, Personal and Ubiquitous Computing 16 (5) (2012) 507–526.

[46] T. H. Dao, S. R. Jeong, H. Ahn, A novel recommendation model of location-based advertising: Context-aware collaborative filtering using ga approach, Expert Systems with Applications 39 (3) (2012) 3731–3739.

[47] L. Hong, L. Zou, C. Zeng, L. Zhang, J. Wang, J. Tian, Context-aware recommendation using role-based trust network, ACM Transactions on Knowledge Discovery from Data (TKDD) 10 (2) (2015) 13.

[48] D. Lee, S. E. Park, M. Kahng, S. Lee, S.-g. Lee, Exploiting contextual information from event logs for personalized recommendation, in: Computer and Information Science 2010, Springer, 2010, pp. 121–139.

[49] Y. S. Lee, X. H. Pham, D. N. Trung, J. J. Jung, H. T. Nguyen, Social context-based movie recommendation: A case study on mymoviehistory, in: Context-Aware Systems and Applications, Springer, 2014, pp. 339–348.

[50] L. He, F. Wu, A time-context-based collaborative filtering algorithm, in: Proc. 2009 IEEE Int. Conf. on Granular Computing, IEEE, 2009, pp. 209–213.

[51] Z. Huang, X. Lu, H. Duan, Context-aware recommendation using rough set model and collaborative filtering, Artificial Intelligence Review 35 (1) (2011) 85–99.

[52] A. Chen, Context-aware collaborative filtering system: Predicting the users preference in the ubiquitous computing environment, in: Location-and Context-Awareness, Springer, 2005, pp. 244–253.

[53] W. Hong, L. Li, T. Li, Product recommendation with temporal dynamics, Expert Systems with Applications 39 (16) (2012) 12398–12406.

[54] H. Bagci, P. Karagoz, Random walk based context-aware activity recommendation for location based social networks, in: Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on, IEEE, 2015, pp. 1–9.

[55] K. Yu, B. Zhang, H. Zhu, H. Cao, J. Tian, Towards personalized context-aware recommendation by mining context logs through topic models, in: Advances in Knowledge Discovery and Data Mining, Springer, 2012, pp. 431–443.

[56] M. Mao, J. Lu, G. Zhang, J. Zhang, Multirelational social recommendations via multigraph ranking, IEEE Transactions on Cybernetics PP (99) (2017) 1–13.

[57] W. Wang, G. Zhang, J. Lu, Member contribution-based group recommender system, Decision Support Systems 87 (2016) 80 – 93.

[58] H. Gao, J. Tang, X. Hu, H. Liu, Exploring temporal effects for location recommendation on location-based social networks, in: Proceedings of the 7th ACM conference on Recommender systems, ACM, 2013, pp. 93–100.

[59] P. He, J. Zhu, Z. Zheng, J. Xu, M. R. Lyu, Location-based hierarchical matrix factorization for web service recommendation, in: 2014 IEEE International Conference on Web Services, ICWS, 2014, 2014, pp. 297–304.

[60] L. Zheng, F. Zhu, S. Huang, J. Xie, Context neighbor recommender: Integrating contexts via neighbors for recommendations, Information Sciences 414 (Supplement C) (2017) 1 – 18.

[61] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, in: Proc. 20th Int. Conf. very large data bases, VLDB, Vol. 1215, 1994, pp. 487–499.

[62] O. Khalid, M. U. S. Khan, S. U. Khan, A. Y. Zomaya, Omnisuggest: A ubiquitous cloud-based context-aware recommendation system for mobile social networks, IEEE Transactions on Services Computing 7 (3) (2014) 401–414.

[63] J. Qi, C. Zhu, Y. Yang, Recommendations based on social relationships in mobile services, Systems Research and Behavioral Science 31 (3) (2014) 424–436.

[64] M. Eirinaki, M. D. Louta, I. Varlamis, A trust-aware system for personalized user recommendations in social networks, IEEE Transactions on Systems, Man, and Cybernetics: Systems 44 (4) (2014) 409–421.

[65] J.-D. Zhang, C.-Y. Chow, Core: Exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations, Information Sciences 293 (2015) 163 – 181.

[66] A. Nocera, D. Ursino, An approach to providing a user of a social folksonomy with recommendations of similar users and potentially interesting resources, Knowledge-Based Systems 24 (8) (2011) 1277–1296.

[67] P. Dwivedi, K. K. Bharadwaj, A fuzzy approach to multidimensional context aware e-learning recommender system, in: Mining Intelligence and Knowledge Exploration, Springer, 2013, pp. 600–610.

[68] N. Zheng, Q. Li, A recommender system based on tag and time information for social tagging systems, Expert Systems with Applications 38 (4) (2011) 4575–4587.

[69] S.-H. Min, I. Han, Detection of the customer time-variant pattern for improving recommender systems, Expert Systems with Applications 28 (2) (2005) 189–199.

[70] N. Hariri, B. Mobasher, R. Burke, Context-aware music recommendation based on latenttopic sequential patterns, in: Proceedings of the sixth ACM conference on Recommender systems, ACM, 2012, pp. 131–138.

[71] P. Symeonidis, E. Tiakas, Y. Manolopoulos, Product recommendation and rating prediction based on multi-modal social networks, in: Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, 2011, pp. 61–68.

[72] H. Wu, K. Yue, X. Liu, Y. Pei, B. Li, Context-aware recommendation via graph-based contextual modeling and post-filtering, International Journal of Distributed Sensor Networks 2015 (2015) 16.

[73] A. Karatzoglou, X. Amatriain, L. Baltrunas, N. Oliver, Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering, in: Proc. 4th ACM Conf. on Recommender systems, ACM, 2010, pp. 79–86.

[74] A. Rettinger, H. Wermser, Y. Huang, V. Tresp, Context-aware tensor decomposition for relation prediction in social networks, Social Network Analysis and Mining 2 (4) (2012) 373–385.

[75] B. Hidasi, D. Tikk, Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2012, pp. 67–82.

[76] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, N. Oliver, Tfmap: optimizing map for top-n context-aware recommendation, in: Proc. 35th ACM SIGIR Int. Conf. on Research and development in information retrieval, ACM, 2012, pp. 155–164.

[77] S. Rendle, Z. Gantner, C. Freudenthaler, L. Schmidt-Thieme, Fast context-aware recommendations with factorization machines, in: Proc. 34th ACM SIGIR Int. Conf. on Research and development in Information Retrieval, ACM, 2011, pp. 635–644.

[78] B. Zou, C. Li, L. Tan, H. Chen, GPUTENSOR: efficient tensor factorization for context-aware recommendations, Information Sciences 299 (2015) 159–177.

[79] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, Cars2: Learning context-aware representations for context-aware recommendations, in: Proc. 23rd ACM Int. Conf. on Information and Knowledge Management, ACM, 2014, pp. 291–300.

[80] C. Zheng, E. Haihong, M. Song, J. Song, Cmptf: Contextual modeling probabilistic tensor factorization for recommender systems, Neurocomputing.

[81] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, Context-aware SVM for context-dependent information recommendation, in: Proc. 7th MDM Int. Conf. on Mobile Data Management, IEEE, 2006, pp. 109–109.

[82] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, H. Kato, Context-aware ranking method for information recommendation, in: Advances in Communication Systems and Electrical Engineering, Springer, 2008, pp. 319–337.

[83] Y. Omori, Y. Nonaka, M. Hasegawa, Design and implementation of a context-aware guide application for mobile users based on machine learning, in: Knowledge-Based and Intelligent Information and Engineering Systems, Springer, 2010, pp. 271–279.

[84] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, S. Yang, Social contextual recommendation, in: Proc. of the 21st ACM Int. Conf. on Information and knowledge management, ACM, 2012, pp. 45–54.

[85] Y. Koren, Collaborative filtering with temporal dynamics, Communications of the ACM 53 (4) (2010) 89–97.

[86] A. Odić, M. Tkalčič, J. F. Tasič, A. Košir, Predicting and detecting the relevant contextual information in a movie-recommender system, Interacting with Computers 25 (1) (2013) 74–90.

[87] M. Unger, A. Bar, B. Shapira, L. Rokach, Towards latent context-aware recommendation systems, Knowledge-Based Systems 104 (2016) 165–178.

[88] P. Do, H. Le, V. T. Nguyen, T. N. Dung, A context-aware collaborative filtering algorithm through identifying similar preference trends in different contextual information, in: Advanced in Computer Science and its Applications, Springer, 2014, pp. 339–344.

[89] K. Ji, R. Sun, X. Li, W. Shu, Improving matrix approximation for recommendation via a clustering-based reconstructive method, Neurocomputing 173 (2016) 912–920.

[90] M. Zhang, J. Tang, X. Zhang, X. Xue, Addressing cold start in recommender systems: A semi-supervised co-training algorithm, in: Proc.

37th ACM SIGIR Int. Conf. on Research & development in information retrieval, ACM, 2014, pp. 73–82.

[91] H. Ma, D. Zhou, C. Liu, M. R. Lyu, I. King, Recommender systems with social regularization, in: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, 2011, pp. 287–296.

[92] X. Liu, K. Aberer, Soco: a social network aided context-aware recommender system, in: Proceedings of the 22nd international conference on World Wide Web, ACM, 2013, pp. 781–802.

[93] J. Liu, C. Wu, W. Liu, Bayesian probabilistic matrix factorization with social relations and item contents for recommendation, Decision Support Systems 55 (3) (2013) 838 – 850.

[94] P. Sitkrongwong, S. Maneeroj, P. Samatthiyadikun, A. Takasu, Bayesian probabilistic model for context-aware recommendations, in: Proc. 17th Int. Conf. on Information Integration and Web-based Applications & Services, ACM, 2015, p. 22.

[95] Z. Xu, L. Chen, G. Chen, Topic based context-aware travel recommendation method exploiting geotagged photos, Neurocomputing 155 (2015) 99–107.

[96] X. Ramirez-Garcia, M. García-Valdez, Post-filtering for a restaurant context-aware recommender system, in: Recent Advances on Hybrid Approaches for Designing Intelligent Systems, Springer, 2014, pp. 695–707.

[97] L. Cui, W. Huang, Q. Yan, F. R. Yu, Z. Wen, N. Lu, A novel context-aware recommendation algorithm with two-level svd in social networks, Future Generation Computer Systems.

[98] Y. Zheng, B. Mobasher, R. Burke, Deviation-based contextual slim recommenders, in: Proc. 23rd ACM Int. Conf. on Information and Knowledge Management, ACM, 2014, pp. 271–280.

[99] G. Chen, L. Chen, Augmenting service recommender systems by incorporating contextual opinions from user reviews, User Modeling and User-Adapted Interaction 25 (3) (2015) 295–329.

[100] D. Yang, D. Zhang, Z. Yu, Z. Wang, A sentiment-enhanced personalized location recommendation system, in: Proceedings of the 24th ACM Conference on Hypertext and Social Media, HT '13, ACM, New York, NY, USA, 2013, pp. 119–128.

[101] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu, S. Yang, Social recommendation with cross-domain transferable knowledge, IEEE Trans. Knowl. Data Eng. 27 (11) (2015) 3084–3097.

[102] Y. Hu, Q. Peng, X. Hu, A time-aware and data sparsity tolerant approach for web service recommendation, in: 2014 IEEE International Conference on Web Services, ICWS, 2014, 2014, pp. 33–40.

[103] W. X. Zhao, S. Li, Y. He, E. Y. Chang, J.-R. Wen, X. Li, Connecting social media to e-commerce: Cold-start product recommendation using microblogging information, IEEE Trans. on Knowl. and Data Eng. 28 (5) (2016) 1147–1159.

[104] A. J. Chaney, D. M. Blei, T. Eliassi-Rad, A probabilistic model for using social networks in personalized item recommendation, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, ACM, New York, NY, USA, 2015, pp. 43–50.

[105] J. Li, C. Chen, H. Chen, C. Tong, Towards context-aware social recommendation via individual trust, Knowledge-Based Systems 127 (Supplement C) (2017) 58 – 66.

[106] X. Ren, M. Song, H. E, J. Song, Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation, Neurocomputing 241 (Supplement C) (2017) 38 – 55.

[107] I. Fernández-Tobías, P. G. Campos, I. Cantador, F. Díez, A contextual modeling approach for model-based recommender systems, in: Advances in Artificial Intelligence, Springer, 2013, pp. 42–51.

[108] S. V. Rodríguez, H. L. Viktor, A personalized location aware multicriteria recommender system based on context-aware user preference models, in: Artificial Intelligence Applications and Innovations, Springer, 2013, pp. 30–39.

[109] C. Lin, R. Xie, X. Guan, L. Li, T. Li, Personalized news recommendation via implicit social experts, Information Sciences 254 (2014) 1–18.

[110] A. Q. de Macedo, L. B. Marinho, R. L. T. Santos, Context-aware event recommendation in event-based social networks, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, 2015, pp. 123–130.

[111] G. Shani, A. Gunawardana, Evaluating recommendation systems, in: Recommender systems handbook, Springer, 2011, pp. 257–297.

[112] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems (TOIS) 22 (1) (2004) 5–53.

[113] R. de Lemos, H. Giese, H. A. Müller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N. M. Villegas, T. Vogel, D. Weyns, L. Baresi, B. Becker, N. Bencomo, Y. Brun, B. Cikic, R. Desmarais, S. Dustdar, G. Engels, K. Geihs, K. M. Göschka, A. Gorla, V. Grassi, P. Inverardi, G. Karsai, J. Kramer, A. Lopes, J. Magee, S. Malek, S. Mankovskii, R. Mirandola, J. Mylopoulos, O. Nierstrasz, M. Pezzè, C. Prehofer, W. Schäfer, R. Schlichting, D. B. Smith, J. P. Sousa, L. Tahvildari, K. Wong, J. Wuttke, Software Engineering for Self-Adaptive Systems: A second Research Roadmap, Vol. 7475, Springer, 2013, pp. 1–32.

[114] M. Litoiu, M. Shaw, G. Tamura, N. M. Villegas, H. A. Müller, H. Giese, R. Rouvoy, E. Rutten, What can control theory teach us about assurances in self-adaptive software systems?, in: R. de Lemos, D. Garlan, C. Ghezzi, H. Giese (Eds.), Software Engineering for Self-Adaptive Systems III, Vol. 9640 of Lecture Notes in Computer Science (LNCS), Springer Berlin Heidelberg, Berlin, Heidelberg, 2017, p. 45, (to appear).

Construction of Recommender System based on Cognitive Model for "Self-Reflection"

Yoshimasa Tawatsuji
Waseda University
Saitama, Japan
y.tawatsuji@aoni.waseda.jp

Yuki Yasuda
Waseda University
Saitama, Japan
gomikuzu.binbin@akane.waseda.jp

Tatsunori Matsui
Waseda University
Saitama, Japan
matsui-t@waseda.jp

ABSTRACT
Every human processes a set of mental schemas for problem solving. We develop and improve these schemas by reflecting on our experiences with errors, which is a type of metacognition (Kayashima, 2008). In this study, we proposed a cognitive model of this "self-reflection" process based on Kayashima's two-layer working memory model, and developed a food recommender system using our cognitive model. In the test simulation, the users were satisfied with the foods that the system recommended, although the recommendation results were unexpected to the users. This implied the system practically worked to satisfy the user's expectation. On the other hand, the candidate recommendations which the system selected as its final output were different from those provided by the users. This suggests that the cognitive model needs improvement in terms of psychological reality.

Author Keywords
self-reflection; cognitive model; recommender system; meta-cognition.

INTRODUCTION
It is important for human beings to acquire problem solving skills to deal with the various problems that they face each day. Metacognition can be defined as the human ability to acquire such problem solving skills [1, 2]. By using metacognition, human beings can observe and improve the problem-solving strategies that they apply. In general, humans avoid committing the same mistakes by improving their strategies. However, it is not clear how people use failure experiences for metacognition and how they improve problem-solving strategies. In this research, we define the refinement of problem-solving skills based on human failure experiences as self-reflection, and attempt to construct a cognitive model expressing the self-reflection process.

HAI 2017, Octobor 17-20, 2017, Bielefeld, Germany
© 2017 ACM ISBN 978-1-4503-5113-3/17/10…$15.00.
DOI: http://dx.doi.org/10.1145/3125739.3132612

Self-reflection is a cognitive activity that is performed internally. We aim to construct a cognitive model expressing the self-reflection process, oriented towards the constructive approach. In this study, we deal the task of food recommendation. Generally, in this domain, it is difficult for the recommender system of a content-based filtering method to make unexpected but satisfying recommendations to the user. This problem is caused by the fact that human preferences are dynamic, even if the system builds a preference model based on the user's history. In this case, the self-reflection function should be able to modify the system to adapt to the user's preferences.

PROPOSED MODEL

Kayashima's model for problem solving: Two Layer Working Memory Model
Many literatures have dealt with metacognition, especially the learning support for metacognitive activities in the field of Artificial Intelligence in Education and Learning Science. Kayashima et al. [3] provided an explicit description for metacognitive activities and proposed the two-layer WM model. This model provides an explanation for the mechanism of cognitive operation (e.g. observation) to cognitive activity, and in this point, this model can be effective to represent the self-reflection process as well.

The model assumed that general problem-solving is a state of transition of the WM influenced by five cognitive activities: "observation," "rehearsal," "evaluation," "virtual application," and "selection." Observation involves carefully looking at the subject and generating the model as the product in the WM. Rehearsal involves keeping the product in the WM to support complicated cognitive operations. Evaluation involves making it possible to search for applicable operators from the knowledge base. The virtual application involves virtually executing an applicable operator in the knowledge base to generate an action list. Selection involves selecting an optimal operator from the results of the virtual application and generating a list of operators to actually apply (action list). When applying the generated action list, the WM's products shifts to new products.

In the WM upper layer, metacognitive activities can be described as cognitive activities that observe and coordinate cognitive activities performed in the WM lower layer. There are two ways to generate products from this metacognitive activity: reflection *in* action and reflection *on* action. The former involves coordinating the cognitive operation in the lower layer and observation thereof in parallel, and generating the observation results in the upper layer as a product. The latter involves observing the products (i.e., cognitive operation process to a product) in the lower layer and generating the process as a product in the upper layer.

Proposed model for problem solving

We attempted to model self-reflection based on the two-layer WM model. Our model focuses on the reflection on action. Previous studies suggest that the failure experience lets humans generate, transpose, and improve strategies [4, 5]. Chiken et al. [6] suggested that strategies are generated by knowledge construction from the failure experience. They also suggested that failure knowledge can be organized into six categories: event, background, process, cause, coping, and summary. In other words, the reflection on action process implies the acquisition of metacognitive knowledge by organizing the failure experience from these categories [7]. Above all, self-reflection is defined as "constructing a strategy by reflection on action through the construction of failure knowledge."

A conceptual model of self-reflection is shown in Figure 1. The model shows that 1) failure feedback triggers the construction of the failure knowledge. 2) To identify the cause product, trial and error of coping (through virtual application) and observation of the result are carried out. 3) As the cause is specified, the background, coping, and cause product are associated with each other. Finally, 4) the association is stored in long-term memory (LTM). The stored strategies in LTM can be easily accessed to adjust cognitive operations in the WM lower layer during the next problem resolution (i.e., reflection in action).

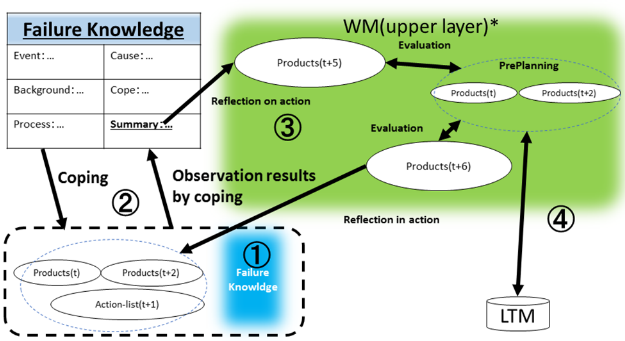


Figure 1. Conceptual model for self-reflection

SYSTEM IMPLEMENTATION

Food recommendation function

The food recommendation function performs cognitive operations in the WM lower layer in Kayashima's model. The problem-solving task of the system can be stated as reducing the number of food recommendation candidates using the cognitive operation module. Then, a food recommendation candidate is regarded as a product of this system. For this system, we built an observation, virtual application, and selection modules.

The observation module observes the conditions of the food recommendation task. It receives context information as input from the user and then selects two contexts that the user might consider important. It's suggested that contextual information should work in recommender systems [8]. The virtual application module narrows down the recommendation candidates, which is the product, based on the observation results. We implemented three methods to determine the food candidates whose context values are 1) close to the user's input values, 2) high, and 3) low. This module generates three types of candidates. The selection module compares each output and then identifies the candidate as next product that meets the user's preference.

The database of meals used for the recommendations of this system was constructed based on a questionnaire survey. One hundred two culinary genres were selected from the most specific culinary genre registered in Tabelog (https://tabelog.com/cat_lst/, 2016/09/01). Fourteen subjects were asked to assign the corresponding context values when deciding what food to eat (e.g. "when you decide to eat pizza, your budget is…") using a 10-point Likert Scale. The context represented the budget, outside temperature he/she felt, number of people to eat with, time passed since he/she woke, and degree of hunger. These context values represent the conditions based on which the food will be chosen. Each food in database had the average context values.

"Self-reflection" function

The self-reflection function is represented by two modules: reflection module and conscious observation module. The reflection module generates and improves strategies based on failure knowledge. The failure knowledge construction begins as the user refuses a recommendation. Specifically, the food recommendation function is repeated. In this case, the combination of all the contexts (here called "virtual-context") is used (i.e., $5 \times 4 = 20$ foods were chosen), and all results are arranged in a list. When the list is displayed to the user, the user selects one food from the list. Then, the module determines the cause product from the selected food, in comparison to recommended food. The cause product can be referred to as the "context of interest" in recommendation or "narrowing down method." If the context of interest is the same as the context of recommended food, the narrowing down method that the system adapted is considered as the cause product of the failure experience.

The conscious observation module adjusts the meal recommendation function based on the strategies constructed in the reflection module. This module works whenever a similar situation as that of a recommendation failure occurs at the start of the recommendation. When the conscious observation module is activated, it displays to the user which part of the internal processing of the food recommendation by the system was changed. For example, the system said "I previously made a mistake in a similar situation. The recom-

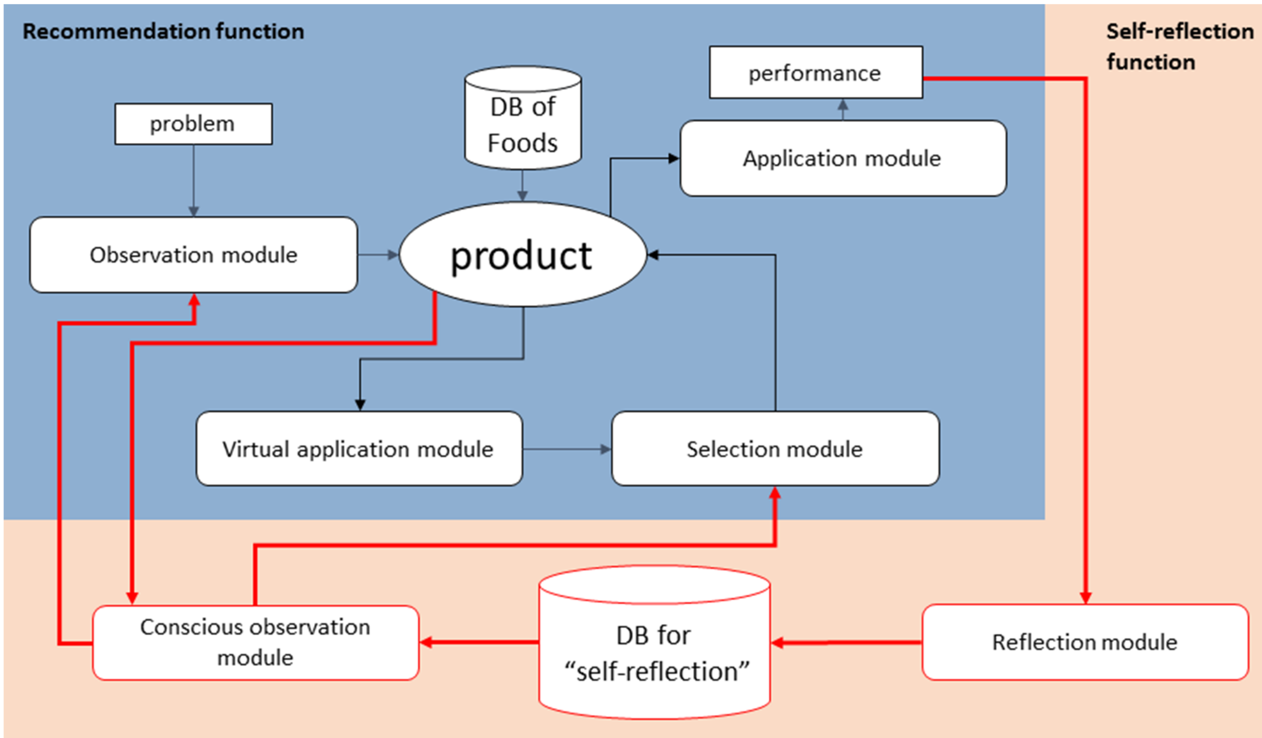


Figure 2. System diagram: The blue and orange part indicate the recommendation and self-reflection function, respectively.

mended food was yakisoba but you preferred yakiniku. *I should have taken into consideration the budget and the number of persons eating with you. Perhaps, the number of persons is important to you! Therefore, in this case, …"*

EXPERIMENT 1

Purpose

The purpose of this experiment is to verify the usefulness of the reflection function and to collect the items recommended by the system for assessing the metacognitive model (ref. Section "Experiment 2").

Procedure

Fifteen subjects (college students, graduate students, and social workers; 10 males, 5 females) participated in the experiment to interact with the system. They were asked to determine their own 10 dietary situations with the given context and received recommendations (10 times for each participant). They decided whether or not to accept the recommendations in each trial. After the interaction, participants were asked to evaluate the degree of 1) satisfaction and 2) unexpectedness of each recommendation with a 10-point Likert scale. In addition, we defined the recommendation acceptance rate to assess the reflective system. We also constructed a general system and a random system to evaluate the verification of the self-reflection module. The general system removes the self-reflection function from the reflective system. On the other hand, the random system decides on all recommendations randomly from the database. Each system also displays the internal processing. For example, the general system presents recom-

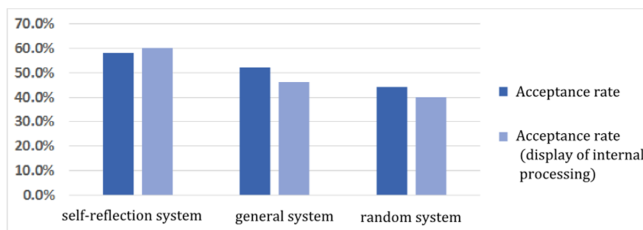


Figure 3. Acceptance rate of each system

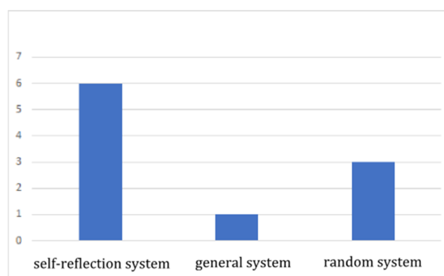


Figure 4. Number of the recommended items whose unexpectedness and satisfaction were evaluated more than 7

mendations such as, *"… This time, I changed the recommendation strategy, that is, I recommend food that everyone, in this case, chooses. So, in this case, …."*

Results

The recommendation acceptance rate for each system is shown in Figure 3. The vertical and the horizontal axes represent the recommendation acceptance rates (%) and each system, respectively. The graphs indicate that the acceptance rate of the reflective system was higher than that of the other systems. This tendency was enhanced when each system displayed the internal processing.

Figure 4 shows the number of recommendations whose satisfactory and unexpectedness were rated as 7 or higher for each system. The graphs indicate that the subjects who interacted with the reflection system tended to highly appreciate the unexpectedness and satisfaction provided by the system recommendations, compared with subjects who interacted with the other systems.

EXPERIMENT 2

Purpose

The purpose of Experiment 2 is to evaluate the validity of the self-reflection model. Because metacognitive activity cannot be observed, we compared the action list of the products (i.e., food candidates) of the system and humans.

Procedure

Three university students (2 males, 1 female) participated in the experiment and were asked to provide food recommendations 10 times to the person with the console (in a chat system). They were instructed to select the foods from the database two times, lining up the candidates and their contexts of interest. We compared the outcomes with the recommendation candidates and contexts of interest generated by the system in Experiment 1.

Results

The contexts of interest that participants reported in their self-reflection were different from that of our system. To assess the reflection module, we applied the participants' contexts of interest to the recommended module to extract the food candidates. Comparisons between the extracted food candidates and user's candidates show that the matching rates between some recommendation candidates on the list rose by as much as 40%. On the other hand, the matching rates for other recommendations decreased.

The characteristic of reflection behaviors presented as the result of an interview are as follows:

- In the second session, he/she seems to consider of his/her budget and number of persons to eat with. Therefore, this time, I recommended foods that were not too expensive and good for sharing with two persons.
- In a similar situation as the second session, I made slightly better recommendations. Hence, I recommended foods based on the degree of hunger. As he/she seems not to mention other context, the budget is taken care of anyway.

DISCUSSION

The participants who interacted with the self-reflection system tended to rate both unexpectedness and satisfaction higher than other recommender systems. The participants also accepted system recommendations, especially those of the self-reflection system. Therefore, our self-reflection system can satisfy their demands.

On the other hand, the results from the Experiment 2 indicate that the candidates generated by the system did not match those generated by participants, which implies that our self-reflection model did not follow the human self-reflection. This result suggests that the high acceptance rate and appreciation of unexpectedness and satisfaction should rely on anything other than the reflection function itself. Experiment 1 showed that the acceptance rate was higher when the system displayed the internal processing to users. Thus, it can be assumed that the display of the system influenced participant's high acceptance rate. As the system behaved as if it was engaged in self-reflection, participants could treat the system as a social being [9]. Such visual performance can help participants accept recommendations.

Another role played by the display of internal processing is encouraging appreciation for unexpectedness and satisfaction. In general, content-based recommender systems have difficulty in recommending the unexpected to users [10]. Our self-reflection system, however, provided such unexpected recommendations. The display contributed to participant's expectation that the system learned his/her preferences. Displaying the internal processing of the reflection system showed which processes and how the system changed based on the failure experience. From this point of view, participants predicted the internal processing model of the system (the rules the system acquired in accordance with their preference). Despite this, when the system recommended foods that were different from what they expected, the unexpectedness was enhanced. When the recommended food was appropriate for the situation they specified, they appreciated the system and were satisfied.

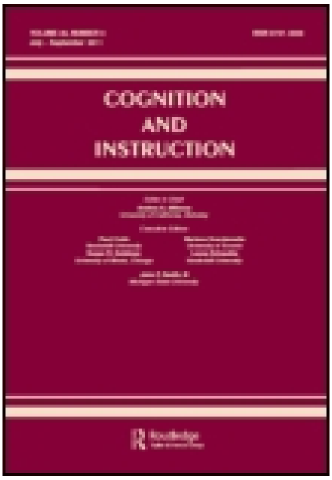
CONCLUSION

In this study, we proposed a cognitive model representing the self-reflection process and implemented it in a food recommender system. We conducted experiments to assess the system and the results indicated that the recommended foods were highly unexpected and satisfying for users. It was also suggested that displaying the internal processing of the system contributed to its high-performance rating from users. The display encouraged participants to believe that the system was learning in accordance with their preferences. On the other hand, the cognitive model was evaluated based on the consistency between food candidates selected by the system and the participants. The results showed that the selected foods were different from each other although both shared the self-reflection process in food recommendation, and that they modified the contexts of interest based on the failure experience.

In future work, we plan to address the following issues. The first is the experimental limitations. In the experiments, it was unclear whether situations participants assumed reflected actual situations. Furthermore, it is also important to study the reflection process in the learning context and consider the metacognitive model to explain the learner's reflection process.

REFERENCES

1. Shinichi Ichikawa. 1996. *Ninchi Shinrigaku (4) Shikou* (Cognitive Psychology (4) Thought). University of Tokyo Press.
2. Giyoo Hatano. 1996. *Ninchi Shinrigaku (5) Gakushu to Hattatsu* (Cognitive Psychology (5) Learning and Development). University of Tokyo Press.
3. Michiko Kayashima, Akiko Inaba and Riichiro Mizoguchi. 2008. A framework of difficulty in metacognitive activity. Transactions of Japanese Society for Information and Systems in Education. 25, 1, 19-31.
4. Tomoya Horiguchi, Isao Imai, Takahito Toumoto and Tsukasa Hirashima. 2008. *Error-based simulation wo mochiita Chugaku Rika no Jugyo Jissen: Newton no daisan housoku wo jirei to shite* (Error-based simulation into classrooms of science in junior high school: A case of Newton's third law). Japan Journal of Educational Technology. 32, supplement, 113-116.
5. Yuri Uesaka. 2009. How learning skills support through cognitive counseling can provide new perspectives in both cognitive research and school curriculum development: Focusing on the strategy of diagram use in problem solving. Cognitive Studies, 16, 3, 313-332.
6. Kunihiko Chiken, Atsuo Hazeyama and Youzou Miyadera. 2005. A programming learning environment focusing on failure knowledge. Transactions on Fundamentals of Electronics, Communications and Computer Science. J88-D-1, 1, 66-75
7. Machiko Sannnomiya. 1996. Shikou ni okeru meta ninchi to chui (Meta cognition and attention in thought), in Shinichi Ichikawa. 1996. *Ninchi Shinrigaku (4) Shikou* (Cognitive Psychology (4) Thought). University of Tokyo Press.
8. Gediminas Adomavicius and Alexander Tuzhilin. (2011). Context-aware recommender systems. Francesco Ricci, et al.(eds). Recommender systems handbook. Springer US. 217-253.
9. Byron Reeves and Clifford Nass. (1996). *The Media Equation: How people treat computers, television, and new media like real people and places*. CSLI Publications and Cambridge.
10. Toshihiro Kamishima. 2007. Algorithms for Recommender Systems (1). Journal of Japanese Society for Artificial Intelligence. 22, 6. 826-837.



Cognition and Instruction
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/hcgi20

Development of a Cognitive-Metacognitive Framework for Protocol Analysis of Mathematical Problem Solving in Small Groups

Alice F. Artz & Eleanor Armour-Thomas

Published online: 14 Dec 2009.

To cite this article: Alice F. Artz & Eleanor Armour-Thomas (1992) Development of a Cognitive-Metacognitive Framework for Protocol Analysis of Mathematical Problem Solving in Small Groups, Cognition and Instruction, 9:2, 137-175, DOI: 10.1207/s1532690xci0902_3

To link to this article: http://dx.doi.org/10.1207/s1532690xci0902_3

COGNITION AND INSTRUCTION, 1992, 9(2), 137–175

Development of a Cognitive-Metacognitive Framework for Protocol Analysis of Mathematical Problem Solving in Small Groups

Alice F. Artzt and Eleanor Armour-Thomas
Department of Secondary Education and Youth Services
Queens College of the City University of New York

A framework is presented that explicitly delineates the roles of metacognition and cognition within small-group heuristic problem solving in mathematics. This framework is used to describe the videotaped behaviors of 27 seventh-grade students of varying ability working in small groups to solve a mathematical problem. The results suggest the importance of metacognitive processes in mathematical problem solving in a small-group setting. A continuous interplay of cognitive and metacognitive behaviors appears to be necessary for successful problem solving and maximum student involvement. Within the groups, students returned several times to such problem-solving episodes as reading, understanding, exploring, analyzing, planning, implementing, and verifying. Stimulated-recall interviews held after completion of the task underscored an additional dimension of importance. Attitudes, particularly those of high-ability students, seemed to affect the interactions and the problem-solving behaviors of fellow group members. The framework shows promise of being a powerful tool for the future study of mathematical problem solving in a small-group setting.

Recently there has been a growing movement toward the use of small groups for mathematics instruction (Crabill, 1990; Johnson & Johnson, 1990; Lindquist, 1989; Noddings, 1989; Rosenbaum, Behounek, Brown, & Burcalow, 1989; Slavin, 1990). Research has shown that, under certain conditions, small-group approaches show positive effects on achievement in mathematics (Davidson, 1985; Davidson & Kroll, 1991). Support for small-group work now comes from the National Council of Teachers of Mathematics (1989) in its publication, *Curriculum*

Requests for reprints should be sent to Alice F. Artzt, Department of Secondary Education and Youth Services, Queens College of the City University of New York, 65–30 Kissena Boulevard, Flushing, NY 11367–1597.

and Evaluation Standards for School Mathematics. The assumption is that through the use of small-group approaches the mathematical problem-solving abilities of students will be improved. Although there is optimism about the efficacy of small-group techniques, proponents acknowledge that little is known about the ways in which the activities used in these small-group approaches produce their positive effects (Bossert, 1988; Slavin, 1989–1990).

Over the last 2 decades, many researchers have studied problem solving in mathematics from a cognitive information-processing perspective. Recent summaries of studies investigating mathematical problem solving (Garofalo & Lester, 1985; Schoenfeld, 1987; Silver, 1987) suggest that a primary source of difficulty in problem solving may lie in students' inabilities to actively monitor and subsequently regulate the cognitive processes engaged in during problem solving. Small problem-solving groups provide natural settings for interpersonal monitoring and regulating of members' goal-directed behaviors. It may be that variables characteristic of these settings are responsible for the positive effects observed in small-group mathematics problem solving. In this exploratory study, we examine the cognitive processing that occurs as individuals engage in mathematical problem solving in small-group settings. Through this investigation, we hope to learn more about how levels of cognitive processes interact and contribute to the successful outcomes of problem solving within small groups.

To examine the interactions between two levels of cognitive processes (i.e., cognitive and metacognitive) observed in the problem-solving behaviors of students working in small groups on mathematics problems, we synthesized a framework for protocol analysis. The procedure has been used to analyze videotapes of students engaged in mathematical problem solving in small-group settings. The framework is derived from research on mathematical problem solving and on cognitive processes discussed in the following three sections.

MATHEMATICAL PROBLEM SOLVING

In mathematics, Polya's (1945) conception of mathematical problem solving as a four-phase heuristic process (understanding, planning, carrying out the plan, and looking back) has served as a standard for investigating problem-solving competence. More recently, Schoenfeld (1983) devised a model for analyzing problem-solving moves that was derived from Polya's. Schoenfeld's model incorporated, within Polya's structure, findings from research on problem solving by information-processing theorists. The model described mathematical problem solving in five episodes: reading, analysis, exploration, planning/implementation, and verification. Garofalo and Lester (1985) built on Polya's and Schoenfeld's structures by developing a framework for analyzing metacognitive aspects of performance on a wider range of mathematical tasks. The four broad component processes—orientation, organization, execution, and verification—are related to

Polya's four phases but are more broadly defined. The four components incorporate Schoenfeld's categories of reading and analysis taken together, planning, implementation, and verification, respectively. Exploration was not specified in the Garofalo and Lester framework. Although Garofalo and Lester indicated the distinctive metacognitive behaviors that may be associated with each category, more research is needed to analyze the specific cognitive processes inherent in mathematical problem solving.

Furthermore, in the application of his framework, Schoenfeld discovered that expert mathematicians returned several times to different heuristic episodes. For example, in one case, an expert engaged in the following sequence of heuristics: read, analyze, plan/implement, verify, analyze, explore, plan/implement, verify. In contrast, the sequence of heuristics for a novice problem solver was just read and explore. More research is needed to examine the sequence of heuristic episodes characteristic of novice problem solvers working in small groups.

COGNITIVE PROCESSES

Current studies of cognitive development focus on cognitive processes as well as on the mechanisms by which development in the processes occurs. Prominent in research on cognitive mechanisms have been strategy selection (e.g., in the solution of computational problems; see Siegler, 1988; Siegler & Shrager, 1984), processing efficiency (e.g., Kail, 1986; Sternberg, 1977), and social scaffolding (e.g., reciprocal teaching; see Palincsar, 1986; Palincsar & Brown, 1984).

Regarding cognitive processes, one lively area of research has focused on the knowledge, monitoring, evaluation, and overseeing that individuals use during any problem-solving endeavor. The term commonly used in the psychological literature for these cognitive processes is *metacognition* (e.g., Brown, 1978; Brown, Bransford, Ferrara, & Campione, 1983; Flavell, 1981; Jacobs & Paris, 1987). For example, Flavell (1981) defined *metacognition* as "knowledge or cognition that takes as its object or regulates any aspect of cognitive endeavor. Its name derives from this 'cognition about cognition' quality" (p. 37). The definition implies that metacognition includes reflection on cognitive activities as well as decisions to modify these activities at any time or place during a given cognitive enterprise. Flavell draws our attention to the dual nature of cognitive processes deployed during any given cognitive enterprise when he stated, "We develop cognitive actions or strategies for *making* cognitive progress and we also develop cognitive actions or strategies for *monitoring* cognitive progress. The two might be thought of as cognitive strategies and metacognitive strategies" (p. 53).

COGNITIVE PROCESSES IN MATHEMATICAL PROBLEM SOLVING

In his framework, Schoenfeld (1983) devised a scheme of parsing protocols into episodes and executive decision points. The executive decision points served as the mechanisms by which the problem-solving process was kept on track. Although his framework focused on the points at which metacognitive decisions may be considered and on their importance in the problem-solving process, in the analyses of protocols, Schoenfeld did not specify the cognitive levels of the episodes themselves.

To examine the problem-solving behaviors and cognitive processes of individuals as they work in small groups, we developed a framework that synthesizes the research on mathematical problem solving with that of cognitive theorists.

DESCRIPTION AND DEVELOPMENT OF FRAMEWORK

The framework for the protocol analysis of problem solving in mathematics developed for this study was designed to differentiate explicitly between cognitive and metacognitive problem-solving behaviors observed within the different episodes of problem solving. Our framework attempts to show a synthesis of the problem-solving steps identified in mathematical research by Garofalo and Lester, Polya, and Schoenfeld, and of cognitive and metacognitive levels of problem-solving behaviors studied within cognitive psychology, in particular, by Flavell (1981).

Schoenfeld's (1985b) framework was used as a foundation in our development. His framework partitioned a problem-solving protocol into "macroscopic chunks of consistent behavior called episodes. An episode is a period of time during which an individual or a problem-solving group is engaged in one large task" (p. 292). The episodes were categorized as *read, analyze, explore, plan/implement*, and *verify*. Through the determination that decisions at the control level would be those that affected the allocation or utilization of problem-solving resources, Schoenfeld allowed for junctures between episodes where these decisions would be most likely to occur. Furthermore, he made specific indications when overt signs of management activity occurred. His framework focused mainly on decision-making behaviors, specifically on statements made about the problem-solving process, at the executive level. A limitation of his framework, Schoenfeld admitted, was that he did not identify statements made about the problem (the more "local" indications of metacognitive behavior). He claimed that, as a result, he was unable to address the important role that consistent monitoring and evaluation of solutions play in the problem-solving process (1985b, p. 293).

Schoenfeld's framework was taken as a starting point; changes were then made to serve the purposes of the present investigation: to delineate explicitly the type and level of cognitive processes individuals use as they work with others in a small-group setting and to understand the mechanisms by which these processes facilitate problem solving. Following Schoenfeld, episodes were used to categorize the behaviors of the individual students within the group. Through the context of the verbal interactions that occurred within the small groups, however, it became clear that several modifications to Schoenfeld's episodes were needed. First, the episode of plan/implement was separated into two distinct episodes. This seemed advisable because the two episodes did not always occur sequentially in the small-group setting. In fact, quite often, a student proposed a plan that was immediately rejected by the other group members. In such cases, no implementation occurred. Second, it became apparent that we had to expand the episodic categories for the coding of student behaviors in groups to include *understanding the problem* and *watching and listening*. The frequent comments students made regarding the conditions of the problem, recognized by Polya as so important in the problem-solving process, served as our reason for including understanding the problem as a distinct episode. Furthermore, the verbal interaction that took place within the small group implied that at certain times students were watching and listening to one another.

Each of the eight problem-solving episodes (read, understand, analyze, explore, plan, implement, verify, watch and listen) was categorized as cognitive or metacognitive. Conceptually, one can distinguish the dual nature of cognitive processing, but operationally the distinction is often blurred. For example, cognition is implicit in any metacognitive activity, and metacognition may be present during a cognitive act, although perhaps not apparent. For this reason, none of the episodes was categorized as purely cognitive or purely metacognitive. The distinction was based on the predominant process observed.

Our working distinction of cognition and metacognition was similar to Garofalo and Lester's (1985, p. 164) description, "Cognition is involved in doing, whereas metacognition is involved in choosing and planning what to do and monitoring what is being done." Metacognitive behaviors can be exhibited by statements made about the problem or statements made about the problem-solving process. Cognitive behaviors can be exhibited by verbal or nonverbal actions that indicate actual processing of information. This distinction between cognitive and metacognitive actions corresponds to those of Flavell (1981) as well. See Table 1 for an outline of the categorization of episodes. A rationale for these categorizations follows.

Episodes of analyzing and planning are, by their very natures, predominantly metacognitive behaviors. Schoenfeld (1985b) stated that:

> In analysis an attempt is made to fully understand a problem, to select an appropriate perspective and reformulate the problem in those terms, and to introduce for

TABLE 1
Framework Episodes Classified by Predominant Cognitive Level

Episode	*Predominant Cognitive Level*
Read	Cognitive
Understand	Metacognitive
Analyze	Metacognitive
Explore	Cognitive and metacognitive
Plan	Metacognitive
Implement	Cognitive and metacognitive
Verify	Cognitive and metacognitive
Watch and listen[a]	

[a]Level not assigned.

> consideration whatever principles or mechanisms might be appropriate. The problem may be simplified or reformulated. (p. 298)

Any statements revealing such thought processes would necessarily be statements made about the problem or about the problem-solving process. Similarly, episodes of planning would be evidenced by statements made about how to proceed in the problem-solving process. Episodes of understanding the problem were categorized as predominantly metacognitive, because this category was assigned only when students made comments that reflected attempts to clarify the meaning of the problem. That is, if a student was making a comment about the meaning of a problem, he or she was also making a comment about the problem. Although it is true that some of the things one does to understand a problem are cognitive, in a coding scheme that relies on the verbal comments of students, it is impossible to decipher the understanding that is being derived during the actual doing of the problem. Behaviors coded as reading were categorized as predominantly cognitive, because they exemplify instances of doing. Behaviors coded as exploring, implementing, and verifying were sometimes categorized as cognitive and sometimes as metacognitive. As Schoenfeld (1987) documented, exploration at the cognitive level alone often results in unchecked "wild goose chases" (p. 210). When exploration is guided by the monitoring of either oneself or one's groupmate, that behavior can be categorized as exploration with monitoring or exploration with metacognition. As a consequence of such monitoring, either self or group regulation of the exploration process can occur, thereby keeping the exploration controlled and focused. The same analysis applies for implementation and verification, which can occur with or without monitoring and regulation. The lack of verbalization during episodes categorized as watching and listening made it impossible to infer a level of cognition. Therefore, these episodes were not categorized as either cognitive or metacognitive. Nonetheless, this last category may still be an important dimension in the process of problem solving in a small-group setting. See the Appendix for a detailed description of the framework.

Figure 1 illustrates the variety of sequences of behavior that could occur during the problem-solving session. Specific examples of the protocol analysis follow the explanation of the mathematical problem.

Unlike previous models, this framework delineates the type and level of processes used as individuals solve mathematical problems in small-group settings. It thereby enables the researcher to examine the role of cognition and metacognition within the heuristic framework of mathematical problem solving in a small-group setting.

METHOD

Subjects

The subjects for this study were 27 seventh-grade students (11 girls, 16 boys) who attended an urban public middle school in the borough of Queens, New York City. The students were selected from three average-ability mathematics classes

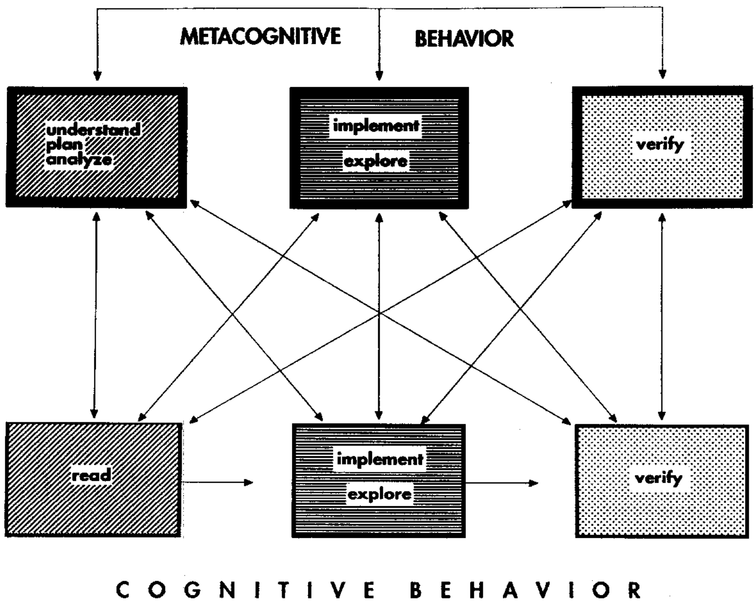


FIGURE 1 A cognitive-metacognitive model. Phases of the problem-solving enterprise.

TABLE 2
Mathematics Problem-Solving Ability:
Percentile Scores on the Metropolitan Achievement Test

	Group					
Students	*1A*	*1B*	*2A*	*2B*	*3A*	*3B*
Quartile (Q)						
Q_4	94	97	99	97	99	94
	82	86	82	86		
	82	77		86		
Q_3	69		73	59	69	63
	50			55	59	55
						50
Q_2			33		25	25
Q_1		18				
Range	44	79	66	42	74	69

taught by two teachers. Only 1 of these students had prior experience in a problem-solving workshop. One week before the study, teachers divided their classes into groups of 4 or 5 students of heterogeneous ability in mathematics. The purpose of this preliminary group experience was to make the students familiar with their group members and the process of group work. In each of the three classes, two groups of students were randomly selected for observation and videotaping, for a total of six groups: Groups 1A, 1B, 2A, 2B, 3A, and 3B. Each group was heterogeneous in mathematical problem-solving ability as measured by the problem-solving section of the Metropolitan Achievement Test. Table 2 lists the students' national percentile scores. The scores are listed according to the groups to which the students belonged and the appropriate quartile in which their scores fell. The groups having the same number (e.g., 1A and 1B) were members of the same class. Each group contained students differing in ability, sex, race, and ethnic background.

Of all six groups, Groups 1A and 2B were the only two whose members all scored above the 50th percentile. In general, these groups were most homogeneous and higher in ability.

Groups 3A and 3B were similar in that they each had only one student who scored in the upper quartile. In fact, each of these students was at the high end of the upper quartile. The students next in ability were in the middle of the third quartile. These groups each contained members at the lowest end of the second quartile. In general, these two groups contained a higher percentage of lower ability students than the other groups, and they had the widest range between the highest ability student and the second highest ability student.

Procedure

Instruments

Students' mathematical problem-solving ability was estimated from their most recent score on the Metropolitan Achievement Test (intermediate form). The test is administered annually in all middle schools. Scores on each student's school record card are recorded as percentiles. Informal interviews were used to obtain mathematics teachers' perceptions of each sample member's ability, attitude, and classroom behavior. Mathematics grades from teachers were a second index of achievement.

Problem Solving in Small Groups

As the students entered the class on the day of the study, they were instructed by the teacher to sit with their groups. They were asked to solve a mathematical problem by working with their group members.

The students were not given a time limit for working on the problem. The teacher called the class together when it was clear that most of the groups had solved the problem and that the few that had not solved the problem seemed unable to proceed further. The problem-solving session lasted between 15 and 20 min.

Videotape of Group Work

In each of the three classes, the investigator (first author) and a research assistant videotaped the two randomly selected groups for the full time they worked on the problem. The videotape was used to provide a permanent record for the coding of problem-solving behaviors. Each of the videotapes was transcribed.

Stimulated-Recall Interview

Within 1 week of the videotaping session, each student participated in a private, stimulated-recall interview as he or she watched the videotape of himself or herself working in the group at six specific times: (a) before his or her group was given the problem, (b) when his or her group was given the problem, (c) when the child began to work on the problem in his or her group, (d) when he or she was deeply involved in working on the problem in his or her group, (e) when the child thought that his or her group had arrived at the solution, and (f) after his or her group had finished working on the problem. These episodes were located by the investigator who viewed each tape and indicated the number on the VCR counter that corresponded to each episode. Students were asked to recall their thoughts during these times in the problem-solving session. The investigator and the research assistant conducted the interviews, which were audiotaped and transcribed.

These interviews provided opportunities to learn about (a) attitudes regarding the task, (b) contributions and participations as group members, and (c) awareness of the problem-solving episodes in which they or their group was engaged (an aspect of metacognition not easily identified through observance of behavior alone).

Coding of Problem-Solving Behaviors

To reach consensus on the categories that best described the observable behaviors, the authors and a research assistant randomly selected one videotape to be used as a pilot tape. The coders worked on the application of the framework until there was strong agreement on how the categories were to be defined and what behaviors were representative of these categories. The interrater reliabilities were high: 93% agreement between the two authors and 91% agreement between the research assistant and one of the authors. In addition, during the actual coding process, if any one of the observers was doubtful about how to categorize a certain behavior, all three observers watched the episode in question and agreed on the appropriate category.

The six videotaped groups consisted of either four or five students. The authors and a research assistant viewed the videotape of each group as it worked on solving the problem. Each viewer watched the behavior of one or two students. The tape was viewed in 1-min intervals, after which each viewer coded the heuristic episode and the cognitive level that best represented the behavior of the student(s) she was observing. Students often exhibited several behaviors during the 1-min time interval. Each behavior was indicated in sequence.

Group Problem-Solving Task

The students were asked to solve the following problem: "A banker must make change of one dollar using 50 coins. She must use at least one quarter, one dime, one nickel, and one penny. How many of each coin must she use to do this?"

The banking problem was selected for several reasons. First, because it cannot be solved using a strict algorithmic procedure, it lends itself to a variety of less structured problem-solving approaches. Second, because students are familiar with money, it was likely that they would understand and be interested in solving the problem. Third, the teachers judged it as an appropriate problem for the ability level of their students.

Before describing the problem-solving behaviors of the students and giving actual protocol examples, we examine the banking problem in light of the proposed framework. We present an outline of several approaches that could be used (many of which the students did use) to solve this problem. These approaches are categorized by episodes—although the episodes are numbered, they are not assumed to be sequential in their occurrence—and cognitive levels as follows:

Episode 1: Reading the problem (cognitive). The student must read or listen to someone else read the problem.
Episode 2: Understanding the problem (metacognitive). The student must understand that there are three conditions that must be met when solving this problem.

1. There must be a total of 50 coins.
2. The value of the coins must be one dollar.
3. There must be at least one quarter, one dime, one nickel, and one penny.

Episode 3: Analyzing the problem (metacognitive). If the students attempt to analyze the problem, there are many ways it can be done. For example:

1. The problem can be reformulated by using the condition that one of each type of coin must be used. That is, four coins (one quarter, one dime, one nickel, and one penny) have the value 25 + 10 + 5 + 1 or 41 cents. This reduces the old problem to a new one having one less condition. That is, now one must find any 46 coins that total 59 cents. No longer is there a restriction about the type of coins that must be used.
2. Because quarters, dimes, and nickels are all multiples of five, their sums, in any combination, will be a multiple of five. Because the sum must be 100, and 100 is a multiple of five, the number of pennies used must also be a multiple of five.
3. For fifty coins to be worth only $1.00, most of the coins selected will have to be pennies.
4. Not many quarters can be used, because four quarters are equivalent to one dollar and 50 coins must be used.

Episode 4: Planning (metacognitive). If the students attempt to plan an approach for solving the problem, there are many ways it can be done. For example:

1. Divide $1.00 into four quarters. Leave one quarter, and keep breaking down the remaining quarters until there are 50 coins.
2. Make a chart using headings of quarter, dime, nickel, and penny. Start with one coin of each type, and then continue adding coins until there are 50 coins totaling $1.00.
3. Start with 50 pennies, and then exchange the pennies for the other coins.
4. Manipulate actual coins to get an idea of how to solve the problem.

Episode 5: Exploring (cognitive and metacognitive). This problem lends itself to a guess-and-test problem-solving approach. This is a form of exploration. If a student is making guesses, testing the guesses, and then making new guesses based on the results of the old ones, he or she is monitoring and regulating the exploration (metacognitive). This, in fact, is an effective technique for solving this particular problem. If, however, the student is merely making a series of random guesses, the student is embarking on an unmonitored exploration (cognitive alone) that is unlikely to result in a solution.

Episode 6: Implementing (cognitive and metacognitive). If a student has devised a plan for solving the problem, he or she is likely to try to implement the plan. If the student does this systematically by monitoring and regulating the implementation (metacognitive), the student is likely to find that the plan either was good and has led him or her toward a solution or was not good and has led him or her to relinquish the implementation and try to devise another plan. If the implementation is not monitored (cognitive alone), however, the student may get buried in the implementation of a poor plan that is unlikely to lead to a solution.

Episode 7: Verifying (cognitive and metacognitive). For an effective verification to take place, the student must be able to take his or her final solution and check that the number of coins is 50, that the total value of the coins is \$1.00, and that one of each type of coin is used. This process entails the ability to add numbers (cognitive) and the ability to monitor the results to check that they meet the conditions of the problem (metacognitive).

Episode 8: Watching and listening (uncategorized). For students to exchange ideas that may facilitate the problem-solving process, they must be willing and able to listen and watch each other.

Protocol Examples

We give several examples of group discussion protocols during different phases of the problem-solving process. The coding of each member's behaviors is given as well. Because the coding was based on the behaviors viewed as well as heard, an overview scenario of the behaviors of the students within the groups is also given. The coding decisions were made on the basis of the overriding behaviors of the students rather than on the basis of each individual statement made.

This first protocol presents the behaviors and the statements of four students (R,O,S,K) in Group 1B during the beginning segment of their problem-solving session (all four students silently read the problem from the blackboard). Student R took the lead in analyzing and devising plans for solving the problem. He was the only one in the group, however, who did practically no writing. All he did was talk about the problem. The other three students did some implementing of his plan while they seemed to be struggling to understand the requirements of the problem. It seemed that they were attempting to solve the problem using Student R's lead before they really understood what the problem asked them to do (evidenced by the statements they made). The protocol follows:

1. S: How many dollars?
2. R: One dollar. We have to use at least one of each to make one dollar. Let's try to get rid of the biggest coins first.

3. S: (Shakes head in agreement and begins to write.)
4. R: Let's get rid of the dimes too so we can get rid of the biggest coins first.
5. K: The dimes?
6. R: Yeah, so you have 35 cents.
7. K: Work with the nickels first.
8. R: No, 'cause we have to use the quarters, dimes, and the nickels.
9. O: Okay, one quarter—(as she writes).
10. R: We have to use a quarter and a dime, which adds up to 35 cents. We have 35 cents already.
11. K: One quarter, one dime (as he writes).
12. O: You know it has to equal one dollar.
13. R: We used two coins already. Let's use five pennies.
14. K: A nickel—
15. R: That's 40 cents.
16. S: Use two quarters.
17. R: No, let's use two of the biggest ones—a quarter and a dime. That's 35 cents.
18. S: We should use pennies then.
19. K: There has to be pennies?
20. S: There has to be 50 coins.
21. K: Fifty coins? One dollar? Okay. (They all start writing.)

Student R. In Statement 2, this student clarifies to himself and others the conditions of the problem (understanding). By deciding to "try to get rid of the biggest coins first," he has immediately launched into a plan. Statements 4, 6, 8, and 10 show that he is sticking to his plan and, in his head, is implementing (cognitive) his plan. By stating that he has accumulated 35 cents (Statements 6 and 10) and by declaring that they have used two coins already (Statement 13), he demonstrates that he is keeping track of or monitoring his implementation (implementing: metacognitive). His suggestion to use five pennies (Statement 13) shows that he is about to go off track from his original plan to use the "biggest coins" first. However, he is kept in line by Student K, who interprets his "five pennies" as a nickel (Statement 14). When Student S suggests the use of two quarters, he rejects it by emphasizing his original plan. Within this segment, Student R's behaviors were coded as follows: reading, understanding, planning, implementing (cognitive and metacognitive).

Student S. In her very first remark in Statement 1, this student shows that she is trying to understand the requirements of the problem (understanding). She shakes her head in agreement to Student R's plan, and, from her subsequent writing, one would assume that she is implementing Student R's plan (implementing: cognitive). Having used the three largest coins, she suggests the use of two quarters (Statement 16). There is no obvious plan at work at this point, and this suggestion seems to mark the beginning of exploring (metacognitive followed by cognitive) of the problem. Student R reminds her of his plan, and it appears from Statement 18 that she, having used all of the largest coins, is ready to use the pennies. In Statement 20, she is again trying to clarify the conditions of the problem for herself and her group. Within this segment, Student S's behaviors were coded as follows: reading, understanding, implementing (cognitive), exploring (metacognitive and cognitive).

Student K. In Statement 5, this student seems to be trying to understand Student R's suggested plan. He then contributes his own plan, which is immediately rejected by Student R (Statement 7). Although his idea of working with the nickels first takes the form of a plan, the fact that it is at such a local level (within Student R's plan), with no apparent rationale behind it, suggests that the appropriate coding is exploring (metacognitive). In Statement 14, he interprets Student R's suggestion to use five pennies as meaning to use one nickel. This monitoring of Student R's implementation helped the group stick to the original plan (implementation: metacognitive). When he asks if pennies must be used (Statement 19), the student shows that he did not yet fully understand the conditions of the problem. During this segment, the behaviors of Student K were coded as follows: reading, exploring (cognitive), implementing (metacognitive), understanding.

Student O. This student was mostly engaged in listening and writing. She only made two comments – once in Statement 9, when she was following Student R's directions, and then again in Statement 12, when she was trying to clarify the conditions of the problem to herself. Within this segment, the behaviors of Student O were coded as follows: reading, watching and listening, implementing (cognitive), understanding.

The second protocol presents the behaviors and the statements of four students (C, D, S, W), referred to as Group 3A, during the middle segment of their problem-solving session. The students have just reached the conclusion that they can reformulate the initial problem by first meeting the conditions of using one of each coin. The students have been working cooperatively, and each student has been engaged in the process of trying to solve the problem. We join them as they try to make sense of where they stand by exploring the revised problem.

1. W: Wait! Wait! We already have 41 cents with 4 coins.
2. C: How much more do we need?
3. W: Now we need 46 coins.
4. S: 46 coins and we need, um–
5. C: 59?
6. W: 59 cents.

7. S: Yeah.
8. D: So use all the pennies.
9. W: Forty-what coins? Forty-six.
10. C: Forty-six coins.
11. W: 59 and 46, what is it?
12. S: No, but you're getting confused 'cause this is the number of cents and this is the number of amount of coins.
13. W: No, but I mean what if we use 46 pennies?
14. D: It's a dollar five.
15. C: Yeah, but we have to use nickels and pennies.
16. D: Yeah.
17. W: All right.
18. S: Maybe we could use 5 nickels and then 41 pennies.
19. W: Try it.
20. S: So 5 nickels is 25 plus 41.
21. C: (Working on his own) Sixty-six again! We already did that. Forty-one plus twenty-five.

Student W. In Statements 1, 3, 6, 9, and 11, this student engages in an assessment of the status of the problem solution. He is figuring out how many coins and how many cents he must have after he already has used 41 cents with 4 different coins. Because we are joining him in the middle of an exploration, this behavior was coded as exploring (metacognitive). In Statement 11, he reveals his confusion of coins and monetary value and, although he does not openly admit it, he is straightened out by student S's comment. In Statements 13, 17, and 19, he makes suggestions and gives encouragement for further exploration of the problem. His suggestion to use 46 pennies was exploratory. Within this segment, the behaviors of Student W were coded as exploring (metacognitive).

Student S. This student's clarifying comment in Statement 12 exemplifies the type of higher level statement that can keep a group's effort on track. In effect, she has monitored the exploration of Student W (exploring: metacognitive). From Statements 18 and 20, it is clear that she has joined in the exploratory efforts of the group by both giving suggestions and making her own calculations (exploring: cognitive and metacognitive). During this segment, the behaviors of Student S were coded as exploring (cognitive and metacognitive).

Student C. During the beginning of this segment, this student was listening to Student W. Afterward, in Statements 2, 5, and 10, he interacted with Student W in exploring the problem. Student C appears to finish the thoughts and sentences that Student W begins (exploring: metacognitive). Although the ideas are not initially his, he adds to the clarification of the issues by helping Student W along. In Statement 15, he reminds the group of the conditions of the problem (although he was incorrect by recalling the need to use both nickels and pennies, because by then the students had already used one of each coin). Whether or not his statement was valid, he was engaging in efforts to clarify the conditions of the problem; thus, his statement was coded as understanding. Finally, in Statement 21, he joins the group in exploring the problem both cognitively (by working on Student S's suggestion) and metacognitively (by recognizing that the result was incorrect and, in fact, one that they had already reached). Within this segment, the behaviors of Student C were coded as follows: watching and listening, understanding, exploring (metacognitive and cognitive).

Student D. During the beginning of this segment, this student was listening to the remarks made by the other group members. In Statement 8, he suggested using "all the pennies." Although his suggestion sounded somewhat arbitrary and was, therefore, coded as exploring (metacognitive) rather than as planning, it had the potential to set the group on the right track. However, instead of adding the 46 pennies to the 41 cents that were already set aside, he added the 46 pennies to the 59 cents that was the sum to be sought (Statement 14). This gave him a total of "a dollar five," with which he could not work. Unfortunately, nobody in the group noticed his error. Within this segment, the behaviors of Student D were coded as follows: watching and listening, exploring (metacognitive and cognitive).

The third protocol presents the behaviors and the statements of four students (D, J, P, T), referred to as Group 2A, and the teacher (G) during the last segment of their problem-solving session. In this group, Student J appeared to be the prime problem solver. Because she did not have a pen or pencil, she dictated her ideas to Student P, who struggled to keep up with her suggestions. Student D busily worked on his own explorations, whereas Student T, the least involved, intermittently reminded fellow group members of the conditions of the problem. We join these students after their initial analysis that they need to find 59 cents using 46 coins. They have been lost in exploration for approximately 7 min.

1. J: Listen, if we have to use one of each, already we have 41 cents. We have 4 coins right? That means we need how many more coins? We need 46 more coins. So 46 coins and 41 cents. We have to break it down into nickels and pennies and everything else.
2. T: Pennies. Use all pennies.
3. J: Yeah, but that's too many. If we use all pennies, we wouldn't have enough.
4. D: It all depends on the pennies [Student T]. I bet you it all depends on the pennies.
5. T: I know. (Student P is holding the pencil, not knowing what to write. Student S does not have a pencil or paper.)
6. J: I'm confused now. (Thinks awhile and then instructs Student P) Put 40 pennies down.

7. P: 40 pennies?
8. J: Yeah—put 40 pennies. Now put one quarter, one dime. So that's 25, 35, (45, 55, 65 to herself) 75 cents. So we need 7 more coins.

(P is writing and trying to calculate to catch up with J's ideas.)

9. J: Use 45.
10. D: (Interrupting J) I got it! I got it! Look, 20 pennies, 1 nickel, 5 dimes, and 25 cents equals a dollar.
11. P: That's only . . . (counting up the number of coins)—
12. T: You've got to use 50 coins.
13. D: Oh, I was so close.
14. K: (Getting back to work with P) All right, so how much do we have here now? We have 44 coins. You got 80 cents. (Calculating to herself) That's 46. We need 4 more coins: 42, 43, 44, 45, 46. No, two. (P looks confused) Add another dime (calculating to herself).
15. D: It would have been easier if they told us how much money do we give the banker.
16. J: Put another dime in.

(T reads the problem to D.)

17. J: (Calculating what P has written) That's 50 coins. Look so that's . . . (calculating to herself) I got it! I got it! Look, look, 45 pennies, 2 nickels, 2 dimes, and a quarter. That's it.
18. D: Yea [Student J]!
19. T: Yes [Student J]!
20. D: Champion! We have it. Yea!
21. P: (Still looking bewildered, trying to figure out what he has written on the paper.)
22. J: (To P) What are you doing? Two nickels. . . .
23. P: Ten cents is two nickels.

(The teacher walks by.)

24. D: Miss G, we think we got it.
25. J: (To Miss G) Forty-five pennies, 2 nickels, 2 dimes, and 1 quarter.
26. G: But how come [Student P] is saying no?
27. P: Four, five, six . . . (Talking out loud as he still tries to figure it out.)
28. D: (To P) Carry the one.
29. P: I did.
30. J: (Pointing to the numbers on P's paper) 45, 55, 65, 75–75 and one quarter is a dollar. We got it.

Student J. In Statement 1, this student is clearly analyzing the problem. Student T comes up with a suggestion that she rebuffs in Statement 3. The student's suggestion was done at the local level and could, therefore, be categorized as a suggestion for exploration. Student J's statement would thus be categorized as monitoring the exploration (exploring: metacognitive). In Statement 6, Student J launches her own exploration by telling Student P to "put 40 pennies down." She calculates and monitors her own suggestion in Statement 8 (exploring: metacognitive and cognitive). By Statement 9, she hits on the correct number of pennies but is interrupted by Student D. When she returns to her ideas in Statement 14, she forgets about her suggestion of using 45 pennies and evaluates the status of the problem with her original idea of using 40 pennies. From this point on, she does most of the work in her head (Student P cannot keep up with her pace). Finally, in Statement 17, she checks her own exploration and discovers that she has solved the problem. She states her answer to the teacher and impatiently tries to help Student P verify the solution. Her final verification is in Statement 30. Her declaration, "We got it," shows that she has verified and checked the solution against the conditions of the problem. She has not merely added numbers but has also monitored the meaning of those numbers. This is an example of verifying at the metacognitive level. Within this segment, the behaviors of Student J were coded as follows: analyzing, exploring (metacognitive and cognitive), verifying (metacognitive and cognitive).

Student P. At this point in the problem-solving session, this student was acting as a secretary. Primarily, he was taking instructions from Student J. At one point, he took a moment to monitor Student D's incorrect proclamation that he had solved the problem (Statement 11). Because Student D's solution came out of extensive exploration, Student P's comment would be categorized as exploring (metacognitive). Because most of his behaviors entailed writing numbers that he did not seem to understand fully, his behavior was coded as exploring (cognitive). At the end of the session, Student P was attempting to verify Student J's solution. This was only at the cognitive level, however, because he still did not seem to have a grip on the problem. Within this segment, the behaviors of Student P were coded as exploring (metacognitive and cognitive).

Student D. This student picked up on Student T's suggestion that the number of pennies used would be very important (Statement 4). This was coded as analyzing. He strayed from the group and went off into his own calculations until he declared his solution in Statement 10. His calculations, compounded by his monitoring of the calculations that led him to believe that he had solved the problem, served as a rationale for coding these behaviors as exploring (metacognitive and cognitive). Students P and T alerted him to the fact that he had satisfied only some of the conditions of the problem. He understood their point and, in Statement 15, stated his wishes for a change in the wording of the problem.

At the end, he cheered for Student J. He watched the calculations that Student P was making as he tried to verify Student J's solution. During this segment, the behaviors of Student D were coded as analyzing, exploring (metacognitive and cognitive), watching and listening.

Student T. This student did not do any calculations during the entire problem-solving session. He spent most of his time watching and listening to the other students. He gave an important suggestion in Statement 2 to "use all pennies." Because this suggestion came at the local level, it was categorized as exploring (metacognitive). After that, his only other statement was to remind Student D of the conditions of the problem. Within this segment, the behaviors of Student T were coded as watching and listening, exploring (metacognitive), understanding.

RESULTS

Results of Problem Solving in Small Groups

The coding for each of the six groups observed was done on charts such as those shown in Figures 2, 3, and 4. The behavior of each student was categorized in two ways: by episode and by cognitive level. The episode or type of problem-solving behavior in which the student was engaged (read, understand, analyze, explore, plan, implement, verify, watch and listen) was recorded in the appropriate row. An asterisk indicates metacognitive behaviors. All other behaviors, except those categorized as watch and listen, were considered to be cognitive. Those categorized as watch and listen were not assigned a cognitive level.

Students' behaviors were coded in 1-min intervals. The time intervals are listed along the bottoms of Figures 2 through 4. Behaviors (episodes) are listed on the vertical axis and are displayed throughout the charts. Students are distinguished from one another by their first initial. For example, in Figure 3, during the first minute, Student C first read the problem, then tried to understand the problem, and then began to do some exploratory work. During Minute 2, Student C watched and listened to what the other students were doing and saying and then resumed exploratory work. Charting each student's behavior in this way yields a picture of each individual's behavior. As an added outcome, a picture of the group's behavior as a whole seems to emerge. (The protocol example of Group 3A can be located in Figure 3 during approximately the fifth to seventh minutes.)

Each figure contains a small table summarizing the behaviors of each student in the group. These behaviors are categorized as metacognitive, cognitive, and watch and listen. By counting the number of metacognitive behaviors coded for one student and dividing it by the total number of behaviors coded in the group, a profile can be obtained of each group member's metacognitive contributions

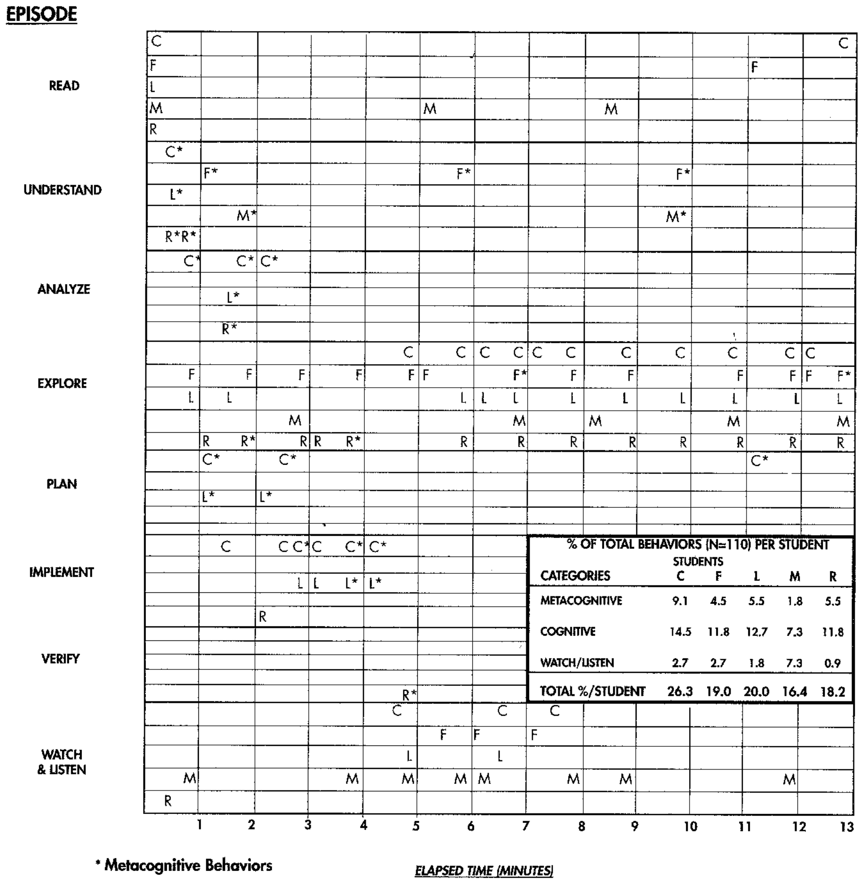


FIGURE 2 Diagram of behaviors in Group 1A.

within the group. The same is done for each member's cognitive behaviors and for watch and listen behaviors.

Metacognitive and Cognitive Behaviors

Table 3 shows the number and percentage of behaviors coded as metacognitive, cognitive, and watch and listen. Of 442 behaviors coded, 38.7% were metacognitive, 36.0% were cognitive, and 25.3% were in the watch and listen category, an undetermined cognitive level.

The metacognitive behaviors as a percentage of the total behaviors coded ranged from a low of 26.3% in Group 1A (the only group that did not solve the problem) to a high of 51.6% in Group 2A. The cognitive behaviors as a percentage of the total behaviors coded ranged from a low of 23.2% in Group 3B to a high

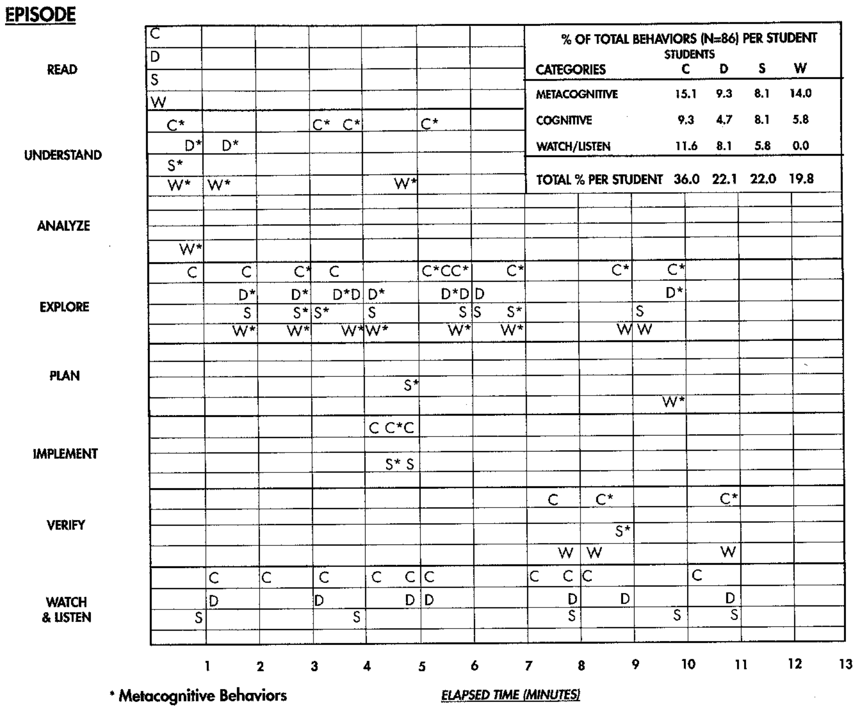


FIGURE 3 Diagram of behaviors in Group 3A.

of 58.2% in Group 1A. The ratio of metacognitive to cognitive behaviors ranged from a low of .45 in Group 1A to a high of 2.00 in Group 1B.

Problem-Solving Episodes and Cognitive Levels

Table 4 lists the percentage of cognitive and metacognitive behaviors coded by category for each group. Across all groups, there were 171 behaviors coded as metacognitive. Of these, 62 were in the category exploring (metacognitive), and 55 were in the category understanding. In other words, the greatest percentage of metacognitive behaviors was in exploring (36.3% of all metacognitive behaviors) and in understanding (32.2% of all metacognitive behaviors). In each of these categories, Group 1A had the lowest percentage of these behaviors—exploring (metacognitive), 3.6%, and understanding, 8.2%.

Across all groups, there were 159 behaviors coded as cognitive. Of these, 96 were in the category exploring (60.4% of all cognitive behaviors), and 38 were in the category reading (23.9% of all cognitive behaviors). In other words, the greatest percentage of cognitive behaviors was in exploring, followed by reading.

Of all the episodes coded, the exploring episode (metacognitive and cognitive together) was coded the greatest percentage of times in each group. The percent-

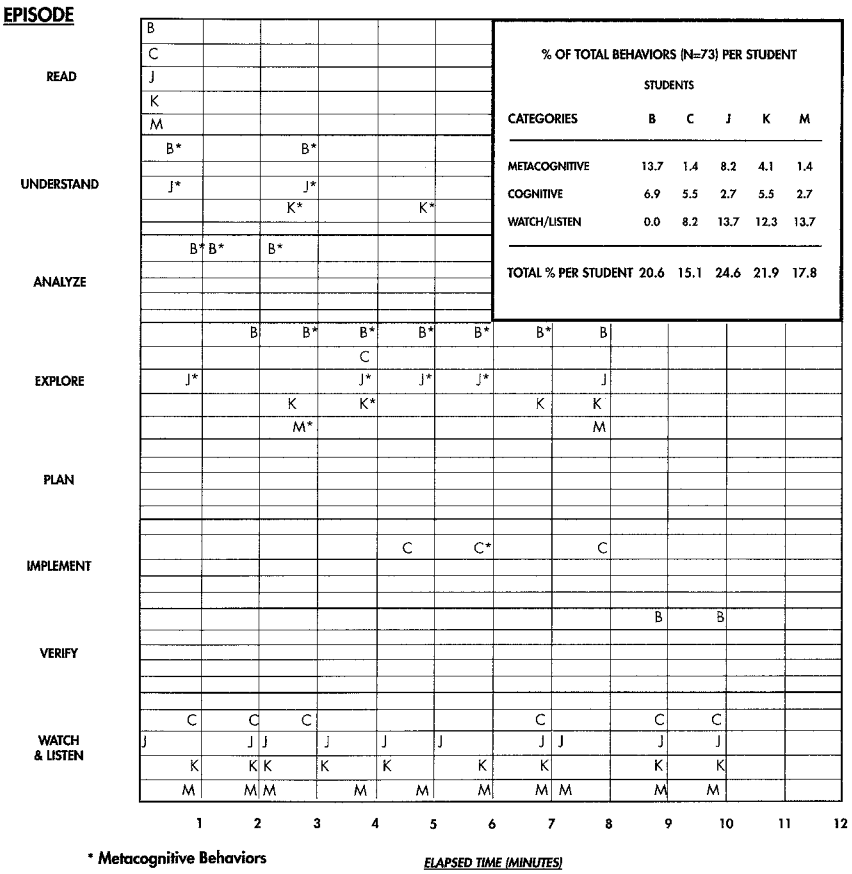


FIGURE 4 Diagram of behaviors in Group 3B.

age of total exploring episodes ranged from 25.3% in Group 1B to 48.0% in Group 1A. That is, in each group, at least one quarter of the episodes coded was in the exploring category.

In total, 112 coded behaviors were in the watch and listen category. Group 3B had the largest percentage of such behaviors (47.9%), and Group 1A had the smallest percentage (15.5%).

We elaborate on these findings in the Discussion section.

Stimulated-Recall Interviews of High-Ability Students

Although all students in the study were interviewed, this article focuses on the interviews of the highest ability student in each group, because they had the greatest potential to solve the mathematical problem. A summary of the narratives that

TABLE 3
Number (and Percentage) of Metacognitive, Cognitive, and Watch-and-Listen Behaviors Per Group

	Group						
Behavior Category	*1A*	*1B*	*2A*	*2B*	*3A*	*3B*	*Total*
Metacognitive	29 (26.3)	32 (47.7)	32 (51.6)	17 (38.7)	40 (46.5)	21 (28.8)	171 (38.7)
Cognitive	64 (58.2)	16 (23.9)	20 (32.2)	18 (40.9)	24 (28.0)	17 (23.2)	159 (36.0)
Watch and listen	17 (15.5)	19 (28.4)	10 (16.1)	9 (20.5)	22 (25.6)	35 (47.9)	112 (25.3)
Total	110 (100.0)	67 (100.0)	62 (100.0)	44 (100.0)	86 (100.0)	73 (100.0)	442 (100.0)

Note. Groups 1A, 2B, and 3B had five members; Groups 1B, 2A, and 3A had four members.

TABLE 4
Percent Distribution of Cognitive, Metacognitive, and Watch-and-Listen Behaviors by Problem-Solving Group

	Group					
Behavior Category	*1A*	*1B*	*2A*	*2B*	*3A*	*3B*
Metacognitive						
Understand problem	8.2	17.9	21.0	11.4	11.6	8.2
Analyze	4.5	8.9	8.1	2.3	1.2	4.1
Explore	3.6	11.9	16.1	15.9	25.6	15.1
Plan	4.5	4.5	4.8	0.0	2.3	0.0
Implement	4.5	3.0	0.0	0.0	2.3	1.4
Verify	0.9	1.5	1.6	9.1	3.5	0.0
Cognitive						
Read	8.2	6.0	12.9	18.2	4.7	6.8
Explore	44.5	13.4	14.5	18.2	15.1	11.0
Implement	5.5	0.0	0.0	0.0	3.5	2.7
Verify	0.0	4.5	4.8	4.5	4.7	2.7
Watch and listen	15.5	28.4	16.1	20.5	25.6	47.9

focused on the students' attitudes about solving the problem in the small-group setting is presented next.

The highest ability members of Groups 2A, 2B, and 3A all revealed insecurities about their own abilities to solve the problem. They expressed their anxieties about the possibility of not being able to solve the problem on their own. They all expressed the desire to receive helpful input from their group members. In contrast, the highest ability members of Groups 1A, 1B, and 3B expressed their desires to work independently. One student revealed her belief that she would proceed more quickly by working alone. She also stated that she was "stubborn" and liked to do things her own way. Another claimed that he preferred to work alone, because that was the way he was "trained" and that is how one is expected to work on an exam. He also intimated that he lacked respect for the abilities of his group members.

The effect of these attitudes on the interactions that occurred within the groups is addressed in the Discussion section.

DISCUSSION

Framework and Cognitive Levels

The framework provides useful information with respect to when, where, and in what frequency group members use processes at the cognitive and metacognitive levels and how these levels of thought may affect the problem solution. It is possible that a certain balance of both cognitive and metacognitive processes within a group is necessary for the problem-solving efforts to result in solution. Indeed, it is interesting that, in this study, the only group that did not solve the problem was the group with the lowest percentage of episodes at the metacognitive level and the highest percentage of episodes at the cognitive level. In fact, in this group, the ratio of metacognitive to cognitive behaviors was lower than any of the ratios of metacognitive to cognitive behaviors of the other five groups. It is also of interest to note that, during the exploratory phase of solving this problem, the unsuccessful group had, by far, the lowest percentage of metacognitive behaviors of all the groups.

Role of Metacognitive Behaviors

The current literature supports the importance of metacognitive behaviors such as active monitoring and subsequent regulation of cognitive processes during the act of problem solving (Baker & Brown, 1982; Flavell & Wellman, 1977; Garofalo & Lester, 1985; Schoenfeld, 1985b; Silver, 1987). When examining the specific instances of these types of metacognitive statements, one gets a better understanding of the ways in which they serve to enhance and propel the problem-solving process.

For example, the statement made by Student R in Group 1B (not reported in the earlier protocol), "We used every coin so far so we don't have to worry about it any more. So we have 41 cents, and we have 46 coins to use. We have to use more pennies," serves to help the group understand the status of the problem solution and the direction in which to go to continue the solution process. Other such statements made by different students were: "No, but you're getting confused 'cause this is the number of cents and this is the number of amount of coins"; and "Listen, if we have to use one of each, already we have 41 cents. We have 4 coins right? That means we need how many more coins? We need 46 more coins. So 46 coins and 41 cents. We have to break it down into nickels and pennies and everything else." Such statements often change the flow of conversation and appropriately redirect the efforts of the group members.

In a different way, the more "local" monitoring statements such as "No, that wouldn't work," "It's gotta be 50 coins," and "Use a lot of pennies" serve to control the group and to keep it from going off on wrong tangents by reminding the group members of the conditions of the problem that must be met and by suggesting the next small steps to take. As revealed by the transcriptions of the videotapes, these statements were made by all students who were caught up in the flow of the problem-solving process.

An example of what happens when there is an absence of consistent monitoring and regulating of the problem-solving process can be seen in Group 1A, which did not solve the problem. Student C declared the incorrect plan of using only nickels and pennies after they had 1 quarter and 1 dime. (The correct solution required 2 dimes.) If her plan had been monitored by another metacognitive statement such as "That won't work" or "Maybe we should try using more dimes also," the group might have had a chance of getting back on track.

These results support the importance of metacognitive processes in mathematical problem solving in small-group settings. The framework developed in this study proved to be an effective tool for capturing the metacognitive behaviors characteristic of effective group work and of effective problem solving.

Role of Cognitive Behaviors

In this study, cognitive activity was evident in all groups. We have seen the important role of metacognitive statements; however, without the presence of students who were able to follow through or implement the metacognitive statements, the problem solving could not have been advanced or completed. For example, in the protocol of Group 3A, the students enacted a plan proposed by one of the group members. After completing the computation, they noticed that their solution was not satisfactory. Through the combined cognitive efforts of performing the calculations and metacognitive efforts of evaluating their solution, the students determined that they had to take a new approach, and thus the problem-solving process was advanced. The interrelationship between metacognitive and cognitive processes is complex, and an appropriate interplay between the two is necessary for successful problem solving to occur.

Role of Watching and Listening

The role of watching and listening is an important variable to consider when studying individuals solving problems in a small-group setting. Watching and listening are as much a part of communication as speaking is, and as Patton, Giffin, and Patton (1989) claimed, "Communication is the essence of the small-group experience" (p. 11).

Although in this study it was not possible to assign a cognitive level to such behavior, watching and listening play a major role in the group process. In fact, the degree of watching and listening behaviors of students may be the defining variable of whether students are engaged in a group interaction at all. For example, in Group 1A the students hardly listened to one another. Perhaps if they had, someone could have helped Student C change her inappropriate plan. Of all six groups, Group 1A had the lowest percentage of watch-and-listen behaviors. The inability of the students to share meanings prevented their group from functioning as a productive unit.

In contrast, most of the students in Group 3B were watching and listening while one person was doing the majority of the work. This is a different version of poor group functioning. Student B assumed a leadership role, and, because of his reported lack of respect for his fellow group members, he dominated the discussion with his own ideas. Research shows that such leadership styles discourage and inhibit the other group members from offering their input (Yukl, 1981). Of all six groups, Group 3B had the highest percentage of watch-and-listen behaviors, none of which was contributed by Student B.

In contrast to the just-mentioned situations, the balance of watching and listening behaviors that occurred in Group 3A contributed to the fruitful interactions that took place. For example, after watching and listening for several minutes, Student D was able to contribute the helpful idea of using mostly pennies. One main advantage of working in a group is that students are able to benefit from group members' ideas. By listening to other people's ideas, one's own ideas are inspired. The extensive degree of interaction that occurred in Group 3A showed that each student was engaged in watching the activities and listening to the ideas of one another.

These examples support the group process theories that show the importance of communication skills for the effective functioning of the group. The balance of watching and listening behaviors during the group problem-solving process is an important issue to be given consideration.

Small-Group Setting

Observing Individuals in a Group Versus Observing a Group

The framework developed for this study was used to observe individual students as they worked in a small-group setting. The presence of group members affects the behaviors of the individual students in various ways and to different degrees. Moreover, in a classroom where the students are arranged in small groups, each group behaves in unique ways. The interactions of individuals working in small groups can be represented on a continuum that ranges from students who (although seated in a group) work independently and do not communicate with others to students who do interact with others in the solution of the problem. Between these two extremes is a multitude of possible scenarios. Figures 5a, 5b, 5c, and 5d depict several situations that can occur. In this study concerning only six groups, we saw a wide range of behavior patterns. For example, Group 1A was most like that depicted in Figures 5a and 5d where the students tended to work independently. Group 3A was a highly interactive group as depicted in Figure 5b, and Group 3B was the "one-man show" as depicted in Figure 5c.

In the literature, the term *group* has proved difficult to define. In fact, according to social psychologist Theodore Newcomb (1951), the term has never achieved a standard meaning. Definitions range from such loose requirements as a group being merely a collection of people (Homans, 1950) to more restrictive definitions that specify size and specific types of within-group communication (Patton et al., 1989). Generally, studies such as this one, which involve decision-making or problem-solving groups, adapt the more restrictive definitions. Patton et al. (1989) listed five conditions necessary for effective group work: (a) two or more people, (b) interdependence, (c) a common goal, (d) communication, and (e)

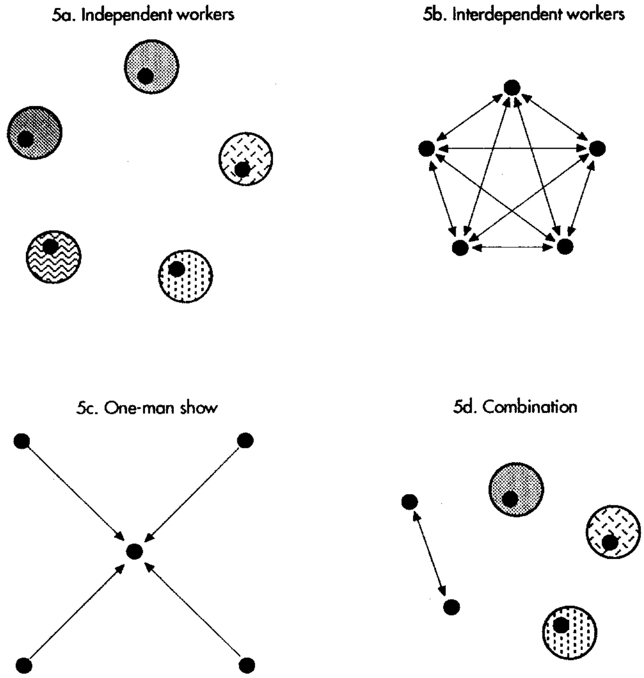


FIGURE 5 Patterns of group interactions.

norms. In the scenario depicted by Figure 5a in which each student works independently, there is little communication and thus ineffective group function. One might then question whether, in fact, this should really be considered a group (in the restrictive sense) or just five individuals seated in a group. At the other extreme, when the involvement of each individual becomes so integrated with the interactions of the other group members, the presence of individuals seems to disappear into the overall tapestry of the group. We have tried to examine the role of cognition and metacognition (as we have defined them) in problem solving in small-group settings by coding the behaviors of individual students. As the group behaviors tend to approach the scenario depicted in Figure 5b on the continuum, however, the behaviors of each of the group members become so interdependent that the group appears to take on its own "collaborative cognition," and the presence of individuals is almost lost.

Cognitive Processes Within a Small Group

The small-group format seems to encourage a spontaneous verbalization that allows the students to externalize their ideas for critical examination. The questioning, elaboration, explanation, and feedback to which these ideas are subjected may be the mechanisms that account for problem solution. Research suggests that the small-grouping schemes that produce the most positive outcomes are structured to maximize these types of behaviors (Slavin, 1989–1990). That is, the most successful grouping strategies are carefully structured to ensure positive interdependence and individual accountability of group members. These structures serve as motivation for students to engage in the active participation within the group that creates an optimum setting for monitoring and regulating behaviors to occur. These are the metacognitive behaviors we have coded in the proposed framework. On this basis, it is reasonable to suspect that the most successful groups, in terms of both solving the problem and getting active involvement of all the group members, should be those with the highest percentages of metacognitive behaviors. In this study, the groups were not structured, although the behaviors of students in Group 3A resembled what would be expected in a group structured for positive interdependence and individual accountability. The results of this exploratory study seem to lend support for the idea that groups having all members actively involved have high incidences of metacognitive behaviors that may be the mechanisms by which problem solving is facilitated.

The diagrams in Figures 2, 3, and 4 represent the behaviors of the students in Groups 1A, 3A, and 3B that are highlighted here because of their contrasting characteristics. A comparison of these charts reveals an observable difference between Group 1A (Figure 2) and the other two groups (Figures 3 and 4). In Groups 3A and 3B, the exploration that occurred was monitored and regulated by group members (as indicated by the asterisks) throughout the duration of the problem-solving session. In Group 1A, there are only a few instances of monitored or regulated exploration. Interestingly, only Group 1A did not solve the problem, and, of all six groups, Group 1A had the lowest percentage of episodes of exploration at the metacognitive level (3.6%; see Table 4). In fact, Group 1A had the lowest total percentage of metacognitive behaviors (26.2%). A related, but not so obvious, difference between Group 1A and the other groups is the smaller percentage of students who were watching or listening to other students (15.5%). The inserted table in Figure 2 clearly shows that none of the students had a high percentage of watching and listening behaviors. In general, one sees this group functioning as individuals who, after the first 3 min, were all engaging in their own private unguided explorations. This is supported by the fact that the highest percentage of each student's individual behaviors was at the cognitive level.

The individualistic, unmonitored explorations of the members of Group 1A contrast with the apparent united effort of the students in Group 3A. (This is indicated in Figure 3 by the presence of all four group members' initials with asterisks in the exploration episodes.) The exploratory activities of this group were characterized by the idea sharing of all group members, accompanied by the subsequent monitoring and regulating of one another's ideas and approaches. The inserted table in Figure 3 shows that the greatest percentage of each student's behavior was metacognitive. This contrasts with the behavior of the individual students in the other groups. From the observers' perspective, it appeared that this metacognitive behavior helped the students discover the solution to which each group member contributed.

Although the members of Group 3B also arrived at the solution, their main problem-solving activities appear to have been dominated by a few individuals (see Figure 4). Specifically, Student B was the driving force in this group. Note that this was the one student who had prior problem-solving experience. The asterisks indicate that Student B was sharing his ideas about the problem with the other students. However, aside from Student B, who did no watching and listening, the highest percentage of behaviors for each of the other students in the group was that of watching and listening. These data reflect the fact that one person dominated this group while the other group members mostly watched and listened.

Ability Levels

Why do groups take on such different behaviors? Why do the proportions of cognitive, metacognitive, and watch-and-listen behaviors differ so greatly among groups? What are the conditions that might bring a group to the highest level of group interaction and group productivity? Research efforts have provided abundant clues to the variables that are likely to impact positively or negatively on a group's performance. High on the list of influential variables are the communication skills, personalities, and intentions of the group members. In this small study, we can begin to investigate further a few ideas. For example, one may wonder how influential are the ability levels and the personalities and attitudes of the group members.

According to the ability data (Table 2), several groups had similar academic profiles (Groups 1A and 2B, Groups 3A and 3B, and Groups 1B and 2A). Despite these similarities, the groups functioned rather differently. Some variables that may have contributed to these differences were the personalities and attitudes of the highest ability member in each group.

For example, the students in Group 1A hardly worked together and were unsuccessful in solving the problem, whereas in Group 2B, aside from one student, the group members were very interactive and succeeded in solving the problem. In Group 1A, the highest ability member was admittedly unwilling to work with the other members of her group. She got fixed on one faulty plan and was not receptive to feedback from her peers. In contrast, in Group 2B, the highest ability member was admittedly insecure about her ability to solve the problem and expressed the desire to receive input from other group members.

A similar contrast existed between the behaviors of Groups 3A and 3B, despite their academic similarities. All the members in Group 3A were highly interactive and cooperative in solving the problem. In contrast, Group 3B was totally dominated by the highest ability member who solved the problem single-handedly and discouraged input from his fellow group members. This student admittedly did not like, or see the purpose of, working with others. He had a competitive attitude and wanted to be the one in his group to solve the problem. Although he was able to solve the problem, he never helped any of his group members and was uninterested in their understanding of the problem. As might be expected, the highest ability member of Group 3A had a very different attitude about working in a group. She expressed a nervousness about the possibility of failing to be able to solve the problem. She said she was anxious to work with her group members in hopes that they would be able to help. Although, in the group, she was the one who made the most high-level metacognitive statements, she was totally influenced by and submerged in the problem solving in which the group was engaged.

In four of the six groups in this study, the highest ability student in each of the groups was primarily responsible for solving the problem. The two groups that did not fit this pattern were Group 1A, which did not solve the problem, and Group 3A, which was so interactive that it was impossible to assign credit to any one person in the group for having solved the problem.

It seems reasonable to conclude that the personalities and attitudes of the highest ability group members have a very powerful effect on the subsequent behaviors of each of the members of the group.

Framework and Heuristic Episodes

The framework was a useful tool for investigating the occurrence and frequency of heuristic episodes. It was evident that the episodes occurred intermittently in all six groups. In all groups, the students returned several times to different episodes. Most often, they returned to the words of the problem to gain a clearer understanding. They could often be heard reminding one another of the conditions that had to be met in the solution of the problem. In fact, all of the groups returned several times to the understanding episode. Most of the groups returned to reading several times. Figure 1 illustrates the ways in which the episodes can occur during a problem-solving session. One would imagine that each time the students returned to an episode they brought new insights with them. So, although they were at the same episode, they were there with a higher level of comprehension.

Exploring (cognitive and metacognitive together) was the behavior that was coded the greatest percentage of times. The exploratory phase, however, was most often accompanied by several other episodes as well. Exploration often led students to have an idea for a plan that, when deemed acceptable by the group members, led to an implementation. Often the implementation was fruitless, and the students returned to their exploration. In several cases, the exploration and analysis of the problem occurred intermittently. The exploration might have allowed the students to gain the familiarity with the problem that is a prerequisite for analysis. In several groups, it appeared that the exploration sparked the analysis, which then sparked further exploration, and then more analysis. This sequence of behaviors looks very similar to the pattern of problem-solving behaviors of the expert mathematician that Schoenfeld (1985b) described in his study.

Framework and Type of Mathematical Problem

There is no question that the type of problem selected for the students to solve was an important variable in this study. Our banking problem was responsible for the high incidence of exploratory behaviors, because it lends itself to a trial-and-error approach. If the problem had been solvable through a more algorithmic approach, there probably would have been more evidence of systematic planning. In general, the type of problem used in this study affected the kinds of problem-solving approaches that might have been observed had a different problem been used. We acknowledge the important influence of context and content on problem-solving behavior, and we admit that the analysis of a single episode is a serious limitation of this study. The main purpose of this research, however, was to find out if the framework used for analyzing the behaviors of individuals as they worked on solving a mathematical problem in a small group showed evidence of being an effective tool. Once establishing that, we intend to apply it to multiple problems.

Study of Thought Processes

Researchers have begun to explore the small-group protocol (as opposed to a single-student or a "think-aloud" protocol) as a vehicle for studying mathematical problem solving, because observers are able to hear the thoughts of the students without interfering in the process (Hart, 1985; Noddings, 1982, 1985; Schoenfeld, 1985a; Silver, 1985). Unfortunately, there still exists a major difficulty in any research that aims to study thought processes. Even in a small group, thoughts are not always verbalized and, therefore, are not easily accessible to the observer. In this study, the only behaviors that could be categorized as metacognitive were those that were audible to the observers. If students were writing silently during episodes of exploration, implementation, and/or verification, their behaviors were categorized as cognitive. It is very possible that students were monitoring their work at these moments and that their metacognitive behaviors were overlooked. It is also possible that silent plans were overlooked as well.

Furthermore, when students work in small groups, they often watch and listen to one another. Because there is no verbalization at these times, it is impossible to assign a cognitive level to this activity. This problem adds an unknown variable to the results and has an effect on the percentages of reported cognitive and metacognitive behaviors. The stimulated-recall interviews were used to gain partial insight into these silent moments.

Compounding the methodological problem of coding was the theoretical complexity of the interplay of cognitive and metacognitive actions in problem solution. One source of the difficulty was identified by Brown (1978) as the difficulty of differentiating metacognitive from cognitive behaviors at both the theoretical and the empirical levels. And, as Sternberg (1985) pointed out, the difficulty is further compounded by the challenge of isolating the relative contributions of both cognitive and metacognitive actions to overall task performance.

CONCLUSION AND IMPLICATIONS

The purpose of this study was to examine the role of cognition and metacognition within the heuristic framework of mathematical problem solving in a small-group setting. A modification of Schoenfeld's framework was developed to delineate explicitly the type and level of processes used as individuals solved mathematical problems in small-group settings. Data analysis suggests the feasibility and usefulness of this framework for research of mathematical problem solving in small groups. With further study, the framework may be of pedagogical use to teachers.

The analysis of problem-solving behavior in the small-group protocols did provide some justification for the differentiation of metacognitive processes from cognitive processes. This is an important distinction with implications at both theoretical and practical levels. Different processes serve different important functions, and future research is needed to gain a better understanding of how interrelationships among processes affect the efficiency and effectiveness of problem solving. The framework also allowed an examination of the occurrence and frequency of heuristic episodes within small groups. The data suggest that, throughout the problem-solving session, students go back and forth using different heuristics intermittently. This behavior seems to play an important role in successful problem solving.

The stimulated-recall interviews provided insight about the attitudes students brought to their groups. Specifically, the attitudes of the high-ability group members manifested themselves in the subsequent behaviors of the group members. These results can provide important information to teachers and researchers who are interested in finding ways to maximize the effectiveness and efficiency of mathematical problem solving in small-group settings.

Most important, the framework shows promise as a powerful tool for the future study of individuals solving mathematical problems in a small-group setting. It can be used to study some of the key questions raised by this study:

1. What is the balance of cognitive, metacognitive, and watch-and-listen behaviors that is most favorable for productive group problem solving?
2. What is the balance of individual group members' contributions that is most favorable for productive group problem solving?

3. When is group problem solving really group problem solving? How should group problem solving be defined?
4. What role does the sequence of heuristic episodes play in the solution of a mathematical problem?
5. What effect do different types of problems have on the heuristic processes used in group problem solving?
6. What is the role of dialogue in small-group settings? More specifically, is there evidence of social scaffolding that occurs naturally in well-structured groups?

Research on questions such as these can begin to be addressed through the application of the framework developed here.

ACKNOWLEDGMENTS

This research could not have taken place without the help of many people. We thank Jamie Rosen for making all of the scheduling arrangements and for helping with the filming and tedious coding. Special thanks go to Gloria Sheskin and Sheryl Garry, to the students and the administrators of Louis Armstrong Middle School for their cooperation and participation in this research, and to Dorothy Feldman and Claire Newman for their editorial help. Thanks also go to Bernard Thomas for his painstaking work on the production of the figures. Most of all, we are grateful to Carol Tittle and Zita Cantwell for their insightful discussions, guidance, advice, editorial assistance, and incredible encouragement. We are especially indebted to Zita Cantwell for endless hours of mentoring that led to the revision of this article. Finally, thanks go to Ann Brown, Alan Schoenfeld, Nel Noddings, and the anonymous reviewers who made such helpful comments.

REFERENCES

Baker, L., & Brown, A. L. (1982). Metacognitive skills in reading. In P. D. Pearson (Ed.), *Handbook of reading research* (pp. 353–394). New York: Longman.

Bossert, S. T. (1988). Cooperative activities in the classroom. In E. Z. Rothkopf (Ed.), *Review of research in education* (Vol. 15, pp. 225–250). Washington, DC: American Educational Research Association.

Brown, A. L. (1978). Knowing when, where, and how to remember: A problem of metacognition. In R. Glaser (Ed.), *Advances in instructional psychology* (Vol. 1, pp. 77–165). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Brown, A. L., Bransford, J. D., Ferrara, R. A., & Campione, J. C. (1983). Learning, remembering, and understanding. In J. H. Flavell & E. M. Markman (Eds.), *Handbook of child psychology: Vol. 3. Cognitive development* (pp. 77–166). New York: Wiley.

Crabill, C. D. (1990). Small-group learning in the secondary mathematics classroom. In N. Davidson (Ed.), *Cooperative learning in mathematics* (pp. 201–227). Menlo Park, CA: Addison-Wesley.

Davidson, N. (1985). Small-group learning and teaching in mathematics: A selective review of the research. In R. Slavin, S. Shara, S. Kagan, R. Hertz-Lazarowitz, C. Webb, & R. Schmuck (Eds.), *Learning to cooperate, cooperating to learn* (pp. 211–230). New York: Plenum.

Davidson, N., & Kroll, D. L. (1991). An overview of research on cooperative learning related to mathematics. *Journal for Research in Mathematics Education, 22,* 362–365.

Flavell, J. H. (1981). Cognitive monitoring. In W. P. Dickson (Ed.), *Children's oral communication skills* (pp. 35–60). New York: Academic.

Flavell, J. H., & Wellman, H. M. (1977). Metamemory. In R. V. Kail & J. W. Hagen (Eds.), *Perspectives on the development of memory and cognition* (pp. 3–33). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Garofalo, J., & Lester, F. K. (1985). Metacognition, cognitive monitoring, and mathematical performance. *Journal for Research in Mathematics Education, 16,* 163–176.

Hart, L. (1985, April). *Factors impeding the formation of a useful representation in mathematical problem solving.* Paper presented at the annual meeting of the American Educational Research Association, Chicago.

Homans, G. C. (1950). *The human group.* New York: Harcourt Brace Jovanovich.

Jacobs, J., & Paris, S. (1987). Children's metacognition about reading: Issues in definition, measurement, and instruction. *Educational Psychologist, 22,* 255–278.

Johnson, D. W., & Johnson, R. T. (1990). Using cooperative learning in math. In N. Davidson (Ed.), *Cooperative learning in mathematics* (pp. 103–124). Menlo Park, CA: Addison-Wesley.

Kail, R. (1986). Sources of age differences in speed of processing. *Child Development, 57,* 969–987.

Lindquist, M. (1989). Mathematics content and small-group instruction in grades 4–6. *Elementary School Journal, 89,* 625–632.

National Council of Teachers of Mathematics. (1989). *Curriculum and evaluation standards for school mathematics.* Reston, VA: The Council.

Newcomb, T. M. (1951). Social psychological theory: Integrating individual and social approaches. In J. H. Rohrer & M. Sherif (Eds.), *Social psychology at the crossroads* (pp. 37–38). New York: Harper & Row.

Noddings, N. (1982, March). *The use of small group protocols in analysis of children's arithmetical problem solving.* Paper presented at the annual meeting of the American Educational Research Association, New York.

Noddings, N. (1985). Small groups as a setting for research on mathematical problem solving. In E. A. Silver (Ed.), *Teaching and learning mathematical problem solving: Multiple research perspectives* (pp. 345–359). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Noddings, N. (1989). Theoretical and practical concerns related to grouping students. *Elementary School Journal, 89,* 607–623.

Palincsar, A. S. (1986). The role of dialogue in providing scaffolded instruction. *Educational Psychologist, 21,* 73–98.

Palincsar, A. S., & Brown, A. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction, 1,* 117–175.

Patton, B. R., Giffin, K., & Patton, E. N. (1989). *Decision-making: Group interaction.* New York: Harper & Row.

Polya, G. (1945). *How to solve it.* Garden City, NY: Doubleday.

Rosenbaum, L., Behounek, K., Brown, L., & Burcalow, J. (1989). Step into problem solving with cooperative learning. *Arithmetic Teacher, 36,* 7–11.

Schoenfeld, A. H. (1983). Episodes and executive decisions in mathematical problem solving. In R. Lesh & M. Landau (Eds.), *Acquisition of mathematics concepts and processes* (pp. 345-395). New York: Academic.

Schoenfeld, A. H. (1985a). Making sense of "out loud" problem-solving protocols. *The Journal of Mathematical Behavior, 4,* 171–191.

Schoenfeld, A. H. (1985b). *Mathematical problem solving.* Orlando, FL: Academic.

Schoenfeld, A. H. (1987). What's all the fuss about metacognition? In A. H. Schoenfeld (Ed.), *Cognitive science and mathematics education* (pp. 189–215). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of Experimental Psychology: General, 117*, 258–275.

Siegler, R. S., & Shrager, J. (1984). Strategy choices in addition and subtraction: How do children know what to do? In C. Sophian (Ed.), *Origins of cognitive skills* (pp. 229–293). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Silver, E. A. (1985). Research on teaching mathematical problem solving: Some underrepresented themes and needed directions. In E. A. Silver (Ed.), *Teaching and learning mathematical problem solving: Multiple research perspectives* (pp. 247–266). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Silver, E. A. (1987). Foundations of cognitive theory and research for mathematics problem-solving instruction. In A. H. Schoenfeld (Ed.), *Cognitive science and mathematics education* (pp. 33–60). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Slavin, R. E. (1989–1990). Research on cooperative learning: Consensus and controversy. *Educational Leadership, 47*(4), 52–54.

Slavin, R. E. (1990). Student team learning in mathematics. In N. Davidson (Ed.), *Cooperative learning in mathematics* (pp. 69–102). Menlo Park, CA: Addison-Wesley.

Sternberg, R. J. (1977). *Intelligence, information processing, and analogical reasoning: The componential analysis of human abilities*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Sternberg, R. J. (1985). *Beyond IQ: A triarchic theory of human intelligence*. New York: Cambridge University Press.

Yukl, G. A. (1981). *Leadership in organizations*. Englewood Cliffs, NJ: Prentice-Hall.

APPENDIX

Cognitive-Metacognitive Framework for Protocol Analysis of Problem Solving in Mathematics

The following framework outlines the interactive relationship between metacognitive and cognitive processes in mathematical problem solving. The episodic categories are described theoretically and empirically. The level or levels of cognition associated with each category are indicated as well. Note that during the course of problem solving these episodes need not occur in the order listed, may occur several times, and may indeed be bypassed completely.

Episode 1: Reading the problem (cognitive)

Description: The student reads the problem.

Indicators: The student is observed as reading the problem or listening to someone else read the problem. The student may be reading the problem silently or aloud to the group.

Episode 2: Understanding the problem (metacognitive)

Description: The student considers domain-specific knowledge that is relevant to the problem. Domain-specific knowledge includes recognition of the linguistic, semantic, and schematic attributes of the problem in his or her own words and represents the problem in a different form.

Indicators: The student may be exhibiting any of the following behaviors: (a) restating the problem in his or her own words; (b) asking for clarification of the meaning of the problem; (c) representing the problem by writing the key facts or by making a diagram or list; (d) reminding himself or herself or others of the requirements of the problem, for example, "Remember, we must use the exact number that is asked for in the problem"; (e) stating or asking himself or herself whether he or she has done a similar problem in the past; and (f) discussing the presence or absence of important pieces of information.

Episode 3: Analyzing the problem (metacognitive)

Description: The student decomposes the problem into its basic elements and examines the implicit or explicit relations between the givens and goals of the problem.

Indicators: The student is engaging in an attempt to simplify or reformulate the problem. An attempt is made to select an appropriate perspective of the problem and to reformulate it in those terms. Examples of statements reflecting that such analysis is occurring are: "After you use all the given information, it becomes an easy problem of addition," and "Because the total is a multiple of five, I think the answer must be divisible by five."

Episode 4: Planning (metacognitive)

Description: The student selects steps for solving the problem and a strategy for combining them that might potentially lead to problem solution if implemented. The student may also select a representation for the information in the problem. In addition, the student may assess the status of the problem solution and make decisions for change if necessary.

Indicators: The student describes an approach that he or she intends to use to solve the problem. This may be in the form of steps to be taken or strategies to be used. Examples of statements that reflect planning include the following: "Let's use the given information first and see what the problem looks like after that"; "Let's work backwards by estimating an answer and see how it must be adjusted to fit the problem"; "Let's draw a chart and fill in the numbers"; "Let's think of a different way to go about this"; and "Let's check back to see where we went wrong."

Episode 5a: Exploring (cognitive)

Description: The student executes a trial-and-error strategy in an attempt to reduce the discrepancy between the givens and the goals.

Indicators: The student engages in a variety of calculations without any apparent structure to the work. There is no visible sequence to the operations performed by the student.

Episode 5b: Exploring (metacognitive)

Description: The student monitors the progress of his or her or others' attempted actions thus far and decides whether to terminate or continue working through the operations. This differs from analysis in that it is less well

structured, and it is further removed from the original problem. If one comes across new information during exploration, he or she may return to analysis in the hope of using that information to better understand the problem.
Indicators: (a) The student draws away from the problem to ask himself or herself or someone else what has been done during the exploration. Examples of such statements are: "What are you doing?" and "What am I doing?" (b) The student gives suggestions to other students about what to try next in the exploration. An example of such a comment is: "It's getting too big; try it with one less." (c) The student evaluates the status of the exploration. Examples of such statements are: "This isn't getting us anywhere," and "I think that's the answer!"

Episode 6a: Implementing (cognitive)

Description: The student executes a strategy that grows out of his or her understanding, analysis, and/or planning decisions and judgments. Unlike exploration, the student's actions are characterized by a quality of systematicity and deliberateness in transforming the givens into the goals of the problem.
Indicators: The student appears to be engaging in a coherent and well-structured series of calculations. There is evidence of an orderly procedure.

Episode 6b: Implementing (metacognitive)

Description: The student engages in the same kind of metacognitive process as in the exploring (metacognitive) phase of problem solving, monitoring the progress of his or her attempted actions. Unlike the exploratory phase, however, the metacognitive decisions build on, check, or revise those previously considered decisions. Furthermore, the student may consider a reallocation of his or her problem-solving resources, given the time constraint within which the problem must be solved.
Indicators: During the implementation phase, the student draws away from the work to see what has been done or where it is leading. The following examples of statements reflect this: "Okay, I used all the given conditions, and now I will start adding what is left"; "Wait. You forgot to use the second point"; and "This is taking too long. Try skipping the odd numbers."

Episode 7a: Verifying (cognitive)

Description: The student evaluates the outcome of the work by checking computational operations.
Indicators: The student redoes the computational operations he or she did before to check that it was done correctly.

Episode 7b: Verifying (metacognitive)

Description: The student evaluates the solution of the problem by judging whether the outcome reflected adequate problem understanding, analysis, planning, and/or implementation. Should the student discover a discrepancy in this comparison search, he or she engages in new decision making for correcting the faulty metacognitive and/or cognitive processing that led to the incorrect solution. The ability to adjust one's thinking on the basis of evaluative information is another indication of self-regulatory competence. Should the evaluation of problem solution indicate an adequacy of or congruence with metacognitive and cognitive processing, the mental reiteration ends.

Indicators: After the student has decided that the solution or part of the solution has been obtained, he or she may review the work in several ways: (a) The student checks the solution process to see whether it makes sense. For example, "When we simplified the problem, did we use all of the given information?" (b) The student checks to see if the solution satisfies the conditions of the problem. For example, "Does our answer satisfy both of the properties that were asked for?" (c) The student explains to a groupmate how the solution was obtained. For example, "I knew it had to be a big number, so I started with the largest numbers first."

Episode 8: Watching and listening (uncategorized)

Description: This category only pertains to students who are working with other people. The student is attending to the ideas and work of others.

Indicators: The student appears to be listening to a group member who is talking or watching a group member who is writing.