Engineering Applications of Artificial Intelligence 104 (2021) 104327

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

CEBRA: A CasE-Based Reasoning Application to recommend banking products

Elena Hernández-Nieves [a,*], Guillermo Hernández [a], Ana B. Gil-González [a], Sara Rodríguez-González [a], Juan M. Corchado [a,b,c,d,e]

[a] Bisite Research Group, University of Salamanca, Spain
[b] Air Institute, Salamanca, Spain
[c] IoT Digital Innovation Hub, Salamanca, Spain
[d] Department of Electronics, Information and Communication, Faculty of Engineering, Osaka Institute of Technology, Osaka, Japan
[e] Pusat Komputeran dan Informatik, Universiti Malaysia Kelantan, Kelantan, Malaysia

ARTICLE INFO

Keywords:
Case-based reasoning
Fog Computing
Virtual agents
Artificial intelligence
Fintech
Commercial banking

ABSTRACT

Following data ethics and respecting the clients' privacy, the banking environment can use the client data that is available to them to offer personalized services to its clients. Intelligent recommender systems can support this attempt through specialized technological architectures. This article proposes the inclusion of CEBRA (CasE-Based Reasoning Application), a case-based reasoning system oriented to commercial banking, in a Fog Computing architecture coordinated by virtual agents. Throughout this article, the model of this architecture is presented and its life cycle is described, and improvements are proposed through the incorporation of several techniques in the retrieve and reuse phases, including the extraction of interests expressed by users on their social network profiles and collaborative filtering systems. A comprehensive case study has been carried out and a dataset of 60,000 cases has been generated to evaluate CEBRA. As a result, the Recommender System is presented, by including, the recommendation algorithm and a REST interface for its use. The recommendations are based on the user's profile, previous ratings and/or additional knowledge such as the user's contextual information. The proposal takes advantage of contextual information to support the promotion of banking and financial products, improving user satisfaction.

1. Introduction

The retail banking process involves the recommendation of products to customers; a service that is important in the development of the Fintech sector and is therefore in a constant process of growth. The acceptance rate of banking products would be greater among customers if the recommendations were targeted at their real needs. Nowadays, any company with a large customer and contact base needs specialised tools to manage data and cross-check information from different databases efficiently. In the case of banks and financial institutions this becomes even more crucial due to the critical nature of the information being handled. These commercial tools, not only serve to build customer loyalty and provide a better service to existing customers, but also improve customer acquisition by impacting the sales process of the banking or financial organisation. In this context there are two different types of technologies that support the customer relationship strategy (Jarrar and Neely, 2002):

1. CRM (Customer Relationship Management). This is a completely internal tool, which provides a faster and more personalised service to current clients. It should be taken into account that CRM solutions adapted to the financial sector usually offer a series of guarantees in terms of legal coverage due to the fact that they have very sensitive and private client data.
2. Customer intelligence. Provides tools for the capture, storage, processing, access, organisation and analysis of customer data. Acquiring new systems, such as predictive modelling systems, involves a considerable investment. These systems build behavioural models to predict response rates, cross-selling opportunities, fraud potentials and credit candidates.

In the research carried out by Melnychenko et al. (2020), there are three stages in the evolution of digital banking. The first stage consists in the introduction of ATMs and call centres in banking activities, which improved customer service. In the second phase, cloud technologies, social networks, analytics and mobile access were applied, allowing

* Corresponding author.
E-mail addresses: elenahn@usal.es (E. Hernández-Nieves), guillehg@usal.es (G. Hernández), abg@usal.es (A.B. Gil-González), srg@usal.es (S. Rodríguez-González), corchado@usal.es (J.M. Corchado).

https://doi.org/10.1016/j.engappai.2021.104327
Received 27 August 2020; Received in revised form 5 April 2021; Accepted 31 May 2021
Available online xxxx

banking institutions to personalise their banking services. And finally the third stage involves the development of digital banking through the use of artificial intelligence, blockchain, programming interfaces, and the robotization of individual business processes. The authors point to a 2019 survey which shows that 76 percent of banks in North America and Europe are seeking to maximise the use of financial technology solutions for payment services and are doing so in order to retain customers in certain segments, while 28 percent of the surveyed companies indicate that they use non-bank suppliers for payments. The survey reveals that almost half of the companies use or are interested in using new payment services, focusing on real-time payments (55 percent), automated clearing houses (44 percent) and the implementation of blockchain services (35 percent).

It is in this context that the application of case-based reasoning techniques is proposed. The recommender systems are designed to provide users with personalized products, powered by automated intelligent mechanisms that enable them to learn from previous users' experiences (Naumov et al., 2019; Nikzad-Khasmakhi et al., 2019). Among the wide range of machine learning technologies, this paper makes use of Case Based Reasoning (CBR) as a paradigm for learning and reasoning through experience. CBR uses automated reasoning that enables to solve new problems through the adaptation and personalization of past solutions. CBR is defined as a problem-solving artificial intelligence technique that can reason from its previous experiences (Aamodt and Plaza, 1994; Chen and Burrell, 2001). It uses its memory to solve new cases, the fact that it does not start from zero distinguishes it from other problem-solving mechanisms (Jubair et al., 2018). CBR is based on different types of similarity metrics and recommends items that meet the specified requirements and the concept of criticism is supported by it (Sridevi et al., 2016).

The motivation behind this research is to advance the researchers' previous study, where a Fog Computing platform was proposed for the recommendation of banking products (Hernandez-Nieves et al., 2020). The use case in the referenced work started with the application of collaborative filtering. Collaborative filtering is a process where product ratings are calculated or estimated using the opinion of different people. It involves users with similar preferences, or products with similar ratings. Therefore, these recommendations are based on other users' ratings of those products or of other similar products. In Hernandez-Nieves et al. (2020) a use case involving the kNN-algorithm was presented. This algorithm predicts a user's rating of a product, taking into account the ratings made by the user's (neighbours) who have made similar ratings for the same products. In this article, we have included this technique in the reusing process of CEBRA as a counterweight to the recommendations obtained by the CBR, so the final recommendation will be the most accurate.

For banking product recommendation, it was decided to develop a CBR instead of a recommender system based on collaborative filtering because it is intended to go beyond a distance rating method, such as the K-Nearest Neighbours. A CBR is able to interact with the environment, assessing its decisions in the real world. In this way, the system judges how good the solutions it proposes are and, in the future, avoids the mistakes it has made before. Even so, it is understood that the quality of the CBR will depend primarily on 5 factors:

1. The ability to understand new situations from previous ones.
2. The initial experience of the system.
3. Of its capacity to adapt.
4. Of its evaluation capacity.
5. Its ability to incorporate new experiences into the case base.

This platform requires a reasoning and a decision making mechanism at a local and global level. That is why this research proposes CEBRA (CasE-Based Reasoning Application). During the search for frameworks we found Colibri and myCBR as general platforms for developing Case-Based Reasoning; in Roth-Berghofer et al. (2012) both are described and compared. The authors explained that the main feature of Colibri is that it reuses previously defined CBR systems and provides a catalogue of already developed systems. myCBR, on the other hand, offers a workstation for the development of knowledge models for CBR systems, such as case structure and similarity measures. Colibri and myCBR are difficult to integrate in a single tool, they are not libraries that can be integrated, that is why it has been decided to contribute to the state of the art by proposing and developing a new integrative framework that could complement the needs of specialized segments, increasing with context application techniques in this segment, such as banking in our case, including the required documentation and instructions to modify the similarity measures in cases where it is required. The framework developed in this article, as an Api/REST, has been written in a readable and ordered code so that any kind of modification and update is possible and therefore it can be reused in other projects. To test the performance of CEBRA, a dataset has been generated with 60,000 cases. Once the case base has been created, CEBRA allows to define a profile for the recommendation, selecting the gender, age, marital status, type of work, etc. resulting in an ordered list of banking products with the highest acceptance rates in the most similar cases.

The article is structured as follows: Section 2 focuses on the virtual agent organization where CEBRA works. Section 3 contains the approach proposed to improve recommendations in the banking sector, and the designed life cycle is described and a case study. Results and discussion are given in Section 4 and the last section of the paper is devoted to the conclusions and future works.

2. Architecture and communication channels

This section describes the aspects related to the Fog Computing architecture and the communication of data between bank divisions, as well as the use of this information to achieve a decision support system for commercial banking. CEBRA is a retail banking advisory agent that operates in a Fog platform, supporting the commercial banking decision process using a combination of local and global decision models and local data. The Fog Computing platform is fully described in Hernandez-Nieves et al. (2020). Fog Computing could be defined as a horizontal architecture at the system-level that distributes computing resources and services, storage, control and networking, at any point in the cloud continuum at the user level (Consortium et al., 2017).

2.1. Virtual agent organizations

In a complex environment it is difficult to determine when and how to recommend banking products effectively. To address this problem from an innovative point of view, this approach takes into account human societies as inspiration. In human societies, it can be found organizational structures that are created and evolve by means of emergent or complex deliberative behaviours. Agent technology may imitate human societies through the constitution of dynamic virtual organizations of agents. These systems are capable of making decisions in an autonomous and flexible way, cooperating with other systems inside an organization (Garcia-Fornes et al., 2011; Oyenan et al., 2009; Rodriguez et al., 2011). Different studies have provided different perspectives on how organizations should be structured in order to adapt themselves easily and efficiently to changes in their environment; adapting old roles to new circumstances or creating new ones (Artikis, 2009; Carrascosa et al., 2009). Agent-based virtual organizations enable the description of structural compositions and functional behaviour, and the inclusion of normative regulations for controlling agent behaviour, for the dynamic entry/exit of components and for the dynamic formation of agent groups (Echeverry et al., 2012). Virtual organizations provide distributed solutions for the resolution of problems, but at the same time they also provide a high degree of autonomy and independence. The development of virtual organizations of agents is still a recent field in the multi-agent system paradigm, it is necessary to develop new methods to model agent-based virtual organizations and innovative techniques to provide advanced organizational abilities to virtual organizations. An analysis of the possibilities and benefits derived from implementing artificial societies shows that virtual organizations are a suitable technology for the complex and highly dynamic operation of Retail Banking. (Fig. 1) shows the proposed virtual organizations adapted to the 3-tier fog-computing architecture.

The agent-based design considered in the proposal provides the system with adaptive capacity and ability to acquire knowledge and make appropriate decisions on the basis of the state of the network. As explained in Chamoso et al. (2018) VAOs offer the system the possibility to develop a flexible core software, with great independence and modularity in the application of recommendation methodologies to provide the best solution. In the architecture hosting CEBRA, the VAOs are located in the Fog layer. These Virtual Organizations of Agents will use their communication and coordination capabilities to share the results obtained by CEBRA. This means that if the agents of the virtual organization in Zaragoza detect that a client in the Valladolid system presents characteristics similar to a success case stored in the Zaragoza CEBRA system, it will communicate with the Valladolid system to share the success case, thus improving the recommendation system. It is considered that in order to manage the system the agents are:

1. Data recovery (DR-VAO): The agents are in charge of communicating and obtaining the data collected by the Fog nodes located in the Fog layer. These agents are connected to the distributed sensors through the middleware.
2. Client (C-VAO): This organization contains CEBRA. This C-VAO recovers and manages the relevant cases. The system first simulates the solutions provided by CEBRA, so that only the successful cases with the highest acceptance rate are chosen. This organization has a customer data agent, so that the bank operator can see the success cases and can interact with them. In this organization, the agents know the preferences of the users.
3. Decision making (DM-VAO): This organization makes its decisions on the basis of information received from DR-VAO and C-VAO. There is an agent to send success cases from one branch to another if a customer similar to those hosted in its database is detected and there is another agent to store cases sent by another branch for its own database. These actions can be carried out by the communication between the agents and the Fog nodes.

This proposal consists in incorporating CEBRA into a Virtual Agent Organisation. This involves providing the system with the possibility of developing a more flexible central software, with great independence and modularity in the application of the recommendation methodologies in order to provide the best solution.

The proposed agent architecture aims to automatically learn new behaviours through the solutions provided by CEBRA. Thus, according to the feedback on the case-based reasoning behaviour, the agent in charge of managing CEBRA improves its efficiency, performance and ability to adapt to the changing environment as new banking product acquisitions are detected.

3. CEBRA: Case-based reasoning model for commercial banking

There are different specializations or varieties of case-based reasoning, especially with regard to representation, indexing or the reasoning mechanisms applied to cases (Aamodt and Plaza, 1994). Usually a CBR system itself is characterized by the case concept. A case is a contextualized piece of knowledge that represents an experience that provides a fundamental lesson to achieve the objectives of the reasoner (Kolodner and Leake, 1996), and must contain a certain level of information and a certain complexity in its internal structure. These types of systems are capable of adapting to different environments or contexts, making generalization possible and allowing for a certain degree of independence from the environment (de Mantaras, 1999). In this research, a CBR could be a suitable recommender system since commercial applications have shown great success with using CBR, mostly because of the advantage that new recommendations (i.e. solutions) can be derived from old recommendations more easily (Skjold and Øynes, 2017).

The action model of a CBR proposed by Riesbeck and Schank (2013) as one of its two fundamental components, is formed by four sequential processes: retrieve, reuse, revise and retain De Mantaras et al. (2005) as well as CEBRA:

1. Retrieve. This is the first stage performed by the CBR system. It is here that case recovery is performed. Two different functions are carried out: access to stored cases and establishing similarity between cases. It is necessary to establish the algorithm of access to the stored cases and the techniques that allow to determine the similarity between the cases.
2. Reuse. In this stage, the most similar cases are received from the previous stage. The aim is to modify and combine or decide which is the most optimal and reuse it.
3. Revise. Verification of the adequacy of the case proposed in the previous stage. Either an expert knowledge system or a human expert is required. The result will be a new case if the solution has been satisfactory or solution repair if it has been incorrect.
4. Retain. It consists in learning from new experiences. The current case and the solution applied to solve it are stored. Efficiencies are assigned to the case. Sometimes it may be necessary to reorganize the case base.

The other fundamental component is the case base or case memory. From this base, the previous solutions are extracted and what has been learned is stored. The case base is in charge of maintaining the representation and organization of the cases. It should take into account the structure of the cases and should try to facilitate, as much as possible, each of the operations in the CBR life cycle.

3.1. Design of CEBRA

The different stages of CEBRA are presented next. In particular, techniques are incorporated to improve the results, which are presented later in the subsubsection CEBRA Retrieve stage and subsubsection CEBRA Reuse stage. K-nearest neighbour classification performance. The life cycle diagram we propose is shown in Fig. 2. At the retrieve stage two acronyms can be observed: Sim_demographic represents the similar cases related with demographic data and socioeconomic indicators store in the database; and Sim_SN represents the similar cases related with opinions and hobbies extracted from Social networks (Twitter).

To build a CBR system a formal framework must be used. To this purpose, we refer to the work of Corchado and Laza (2003), Corchado et al. (2004) where an analytical formalism is established. The authors provided a notation for the aptitudinal components:

Definition 1. A set of case bases (β). A case base $B \in \beta$ is a finite set of cases that is indexed. A case base is defined as a tuple: $(\{c_1, c_2, \ldots, c_n\}, \iota)$.

$\{c_1, c_2, \ldots, c_n\}$ are the cases that conform the case base and ι is the finite set of characteristics that allows the cases to be indexed.

Definition 2. A case (c) represents a past experience. A case is represented by a sequence of environmental states:

$c = \{\text{start\_state}, \{\text{action} \times [\text{intermediate\_state}]\}^{+}, \text{end\_state}\}$. Each state is represented by a set of attributes that define the environment in which the CBR system is located. The states are divided into three groups:

(1) Set of initial states (ini_state), representing the description of the problem to be solved
(2) Set of intermediate states (intern_state), which describe the different states the environment goes through before reaching the final state

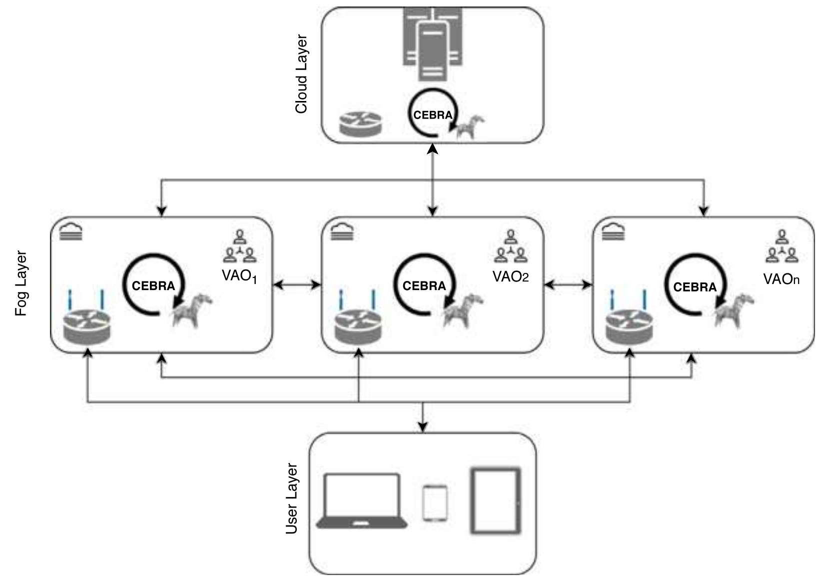

Fig. 1. Fog Computing Architecture hosting CEBRA. Each VAO corresponds to a Fog defined in a certain geographic location (e.g., an area of city), where CEBRA is hosted. CEBRA is also located in the Cloud Layer intended to work with larger datasets.

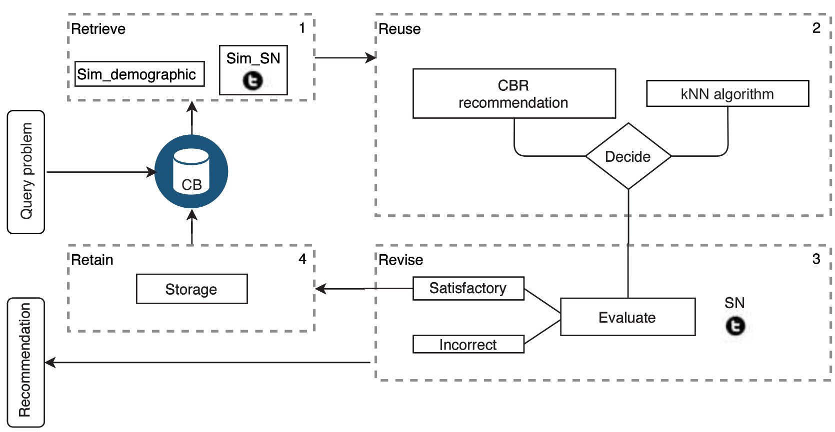

Fig. 2. CEBRA life cycle.

(3) Set of final states ($final\_state$), representing the description of the environment once the initial objectives have been achieved

As well as states a case contains actions, which represent the set of actions applied to each of the states. They are defined by a name and a set of arguments.

Definition 3. A finite set of attributes (k) is a set of properties that allow a state to be described.

Definition 4. An index set (I) is a set of characteristics of $\imath$, with $\imath$ included in k

Definition 5. A set of similarity functions (A) allows to determine the degree of similarity between a problem to be solved and a case.

3.2. Case study of CEBRA

In commercial banking, in the same way as in any other form of commerce, products attempt to satisfy client demand. A bank agency is considered for the sale of banking products such as mortgages, loans, investment products, insurances, etc. We start with an introduction of the attributes and their importance at this context. We have relied on the methodology described by Richter, M. M., et al. in Richter and Weber (2016) to define the phases to build the CBR system. The steps to consider are:

(1) Identify an adequate high-level case scenario.
(2) Introduce characteristics and types that will be applicable to the requested mortgages for the chosen class.
(3) Define importance of attributes for the client class to be weighted.
(4) Define similarity measures at the local level.
(5) Define appropriate finishing and adaptation rules as needed.
(6) Build a case base.

3.2.1. CEBRA retrieve stage

A case study built using a case base with 41 attributes is presented in this section. The case attributes considered by the CBR, which are shown later in this example, have been obtained from a source that we cannot disclose for confidentiality reasons. The dictionary consists of 606 tables organized in 30 categories, each of which is composed of another subset. Within these subsets of tables, the number of sub-tables differs. For example, table number 1 “assets” is composed of 47 sub-tables, while table number 2 “auxiliary” is composed of 254 sub-tables. The selection of case attributes within the dictionary can be represented

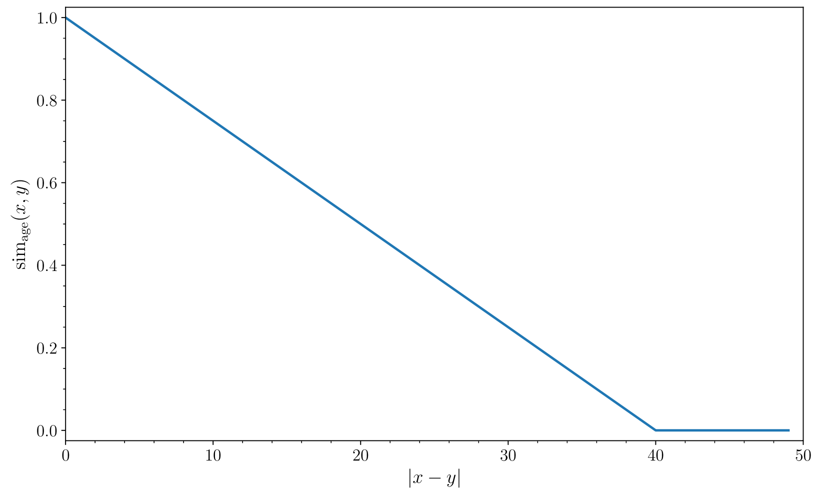

Fig. 3. Age similarity function.

Table 1
CEBRA case attributes in the database dictionary.

Category	Subsets	Attributes
Assets	Person	Age, Gender, Civil status, City, Children, Work, Type of contract, No. of houses owned, No. of cars owned, No. other types vehicles owned
Auxiliary	Product	Applied for a mortgage, Mortgage granted, Early mortgage payback, First time doing early mortgage payback, First time applying for a mortgage, Applied for a loan, Loan granted, First time applying for a Loan, Applied for a deposit, First time applying for a deposit
Accounting	Type of account Monthly balance	Current account, Payroll accounting, Savings account Positive end-of-month balance
Funds	Contracts	Applied for an investment fund, First time applying for an investment fund
Non-compliance	Balance statement	Has there ever been a default?, Regular defaults, Casual defaults
Insurance	Product	Applied for a home insurance, First time applying for a home insurance, Applied for an accident insurance, First time applying for an accident insurance, Applied for a life insurance, First time applying for a life insurance, Applied for a health insurance, First time applying for a health insurance, Applied for a car/motorcycle insurance, First time applying for a car/motorcycle insurance

in this way:

$$\begin{aligned} \mathrm{ATR} = \{&T\_1(Sub42) + T\_2(Sub250) + T\_6(Sub8, Sub12) \\ &+ T\_11(Sub6) + T\_14(Sub5) + T\_25(Sub10)\} \end{aligned} \tag{1}$$

Table 1 is also presented as a summary of the attributes considered for the construction of the case. It is presented in three columns: the first column corresponds to the categories that have been selected within the dictionary of 606 tables mentioned above, the second column shows the subset chosen within the previous category, and finally, the third column shows the attributes that make up a case in CEBRA. In addition to the case attributes considered, the interests expressed by users on their social network profiles are included in the Retrieve Stage (Table 2). These interests will be extracted by applying classifiers to the textual content. Although extracted in the first phase of the CBR cycle, the extracted interest will not be used in this phase, it will be provided to the expert reviewers to complete the case. This reasoning will be explained in more detail in the Revise stage.

Calculation of the similarity functions. Once all the case attributes have been covered, the Query problem has also been considered (Table 3). A random profile has been generated by answering the case attributes (k). A comparison between the query problem (ini_stateQ) and various description of problems to be solved (ini_state1), (ini_state2), (ini_state3), (ini_state4), (ini_state5) are presented below.

Each case attribute requires its own similarity function. As a general rule, a similarity value of 1 is given when the values of two attributes are the same; and a similarity value of 0 when the values are not the same. In this sense, for two values that are different from each other but can be considered moderately similar, a value of 0.5 can be used (Richter and Weber, 2016). This section describes the similarity values that are the most difficult to define, i.e. that they differ from 0 and 1. In order to define similarity functions for each of the attributes, in the case of quantitative numbers such as age, an analytical form is constructed for each of these:

Table 2
Sample client interests that can be extracted by CEBRA from social networks in retrieve stage.

Categories	Attributes
Entertainment	Movies, TV, Radio, Music festivals and concerts, Theatre and musicals
Leisure time	Travels, Sports
Automotive	Cars, Motorcycles, F1, Grand prices, Dakar, Moto GP
Global marketplace	Companies, Brands, Products
Food and drink	Restaurants, How to cook?, MICHELIN Guide, haute cuisine
Gaming	New games, Game console
Health	Healthy food, Diet, Vegan, Yoga, Wearable
Style and fashion	Clothes, Clothing brands, Trends, Fashion magazines
Style and fashion	Furniture, House renovations, Gardening, Decoration trends

(i) Age. A linear relationship is established with the absolute value of the age difference, 0 is the similarity in cases where the age is 40. In cases in which the difference is greater, the similarity continues to be zero.

Table 3
Case attributes in query problem.

ini_stateQ	*k*	*ini_state1*	*ini_state2*	*ini_state3*	*ini_state4*	*ini_state5*
34	Age	45	68	25	38	74
F	Gender	M	M	M	F	M
M	Civil status	M	M	S	M	W
Sa	City	Sa	Sa	Sa	Sa	Sa
1	Children	2	0	0	2	3
Y	Work	Y	R	Y	Y	R
T	Type of contract	G. O.	P	T	P. C.	P
1	No. of houses owned	1	1	0	0	1
0	No. of cars owned	1	1	0	1	0
0	No. other types vehicles owned	0	0	1	0	0
Y	Current account	Y	Y	Y	Y	Y
Y	Payroll accounting	Y	Y	Y	Y	Y
N	Has there ever been a default?	N	Y	Y	N	Y
N	Regular defaults	N	N	N	N	N
N	Casual defaults	N	Y	Y	N	Y
U	Positive end-of-month balance (usual, medium, low)	U	L	M	U	U
N	Saving account	N	Y	N	N	Y
Y	Debit card	Y	Y	Y	Y	Y
Y	Credit card	Y	N	N	Y	Y
Y	Applied for a mortgage	N	N	N	Y	N
Y	Mortgage granted	n/a	n/a	n/a	Y	n/a
N	Early mortgage payback	n/a	n/a	n/a	N	n/a
n/a	First time doing early mortgage payback	n/a	n/a	n/a	n/a	n/a
Y	First time applying for a mortgage	n/a	n/a	n/a	Y	n/a
N	Applied for a loan	Y	N	Y	N	Y
n/a	Loan granted	Y	n/a	Y	n/a	Y
n/a	First time applying for a Loan	Y	n/a	Y	n/a	N
N	Applied for a deposit	N	N	N	N	Y
n/a	First time applying for a deposit	n/a	n/a	n/a	n/a	Y
N	Applied for an investment fund	N	Y	N	N	Y
n/a	First time applying for an investment fund	n/a	N	n/a	n/a	N
Y	Applied for a home insurance	Y	N	N	Y	N
Y	First time applying for a home insurance	Y	n/a	n/a	Y	n/a
N	Applied for an accident insurance	N	N	N	N	N
n/a	First time applying for an accident insurance	n/a	n/a	n/a	n/a	n/a
Y	Applied for a life insurance	Y	N	N	Y	N
Y	First time applying for a life insurance	Y	n/a	n/a	Y	n/a
N	Applied for a health insurance	N	N	N	Y	N
n/a	First time applying for a health insurance	n/a	n/a	n/a	Y	n/a
N	Applied for a car/motorcycle insurance	N	N	Y	N	N
n/a	First time applying for a car/motorcycle insurance	n/a	n/a	Y	n/a	n/a

Gender: F (Female), M (Male).
Civil status: M (Married), S (Single), W (Widower).
City: Sa (Salamanca, Spain).
Type of contract: G. O. (Government Official), P. C. (Permanent Contract), P (Pension), T (Temporary).
Positive end-of-month account balance: U (Usual), M (Medium), L (Low).

Formally:

$$\mathrm{sim}_{\mathrm{age}}(x, y) = \max\left\{1 - \frac{|x-y|}{40}, 0\right\} \quad (2)$$

The behaviour of this function is shown in Fig. 3. Note that, although more complicated models could be defined to more accurately capture the differences between ages, the uniparametric model presented here has been chosen for simplicity. In future revisions there would be room for modelling a more complicated similarity function, provided that it is supported by experimental data.

(ii) Children. The relation of similarity is established in such a way that for a value of 0 children against 0 children a value of 1 is given, for a value of 1 against 0 children a value of 0.5 is given; for a value of two children against a value of 0 a value of 0.13 is given; etc.

$$\mathrm{sim}_{\mathrm{children}}(x, y) = \left(\frac{1}{2}\right)^{|x-y|}. \quad (3)$$

The motivation behind this expression is to model a rapid geometric decay with the differences in the values of the attribute. Expressed as a matrix it would be

$$\mathrm{sim}_{\mathrm{children}}(x, y) = \begin{pmatrix} 1 & 0.5 & 0.25 & 0.13 & \dots \\ 0.5 & 1 & 0.5 & 0.25 & \dots \\ 0.25 & 0.5 & 1 & 0.5 & \dots \\ 0.13 & 0.25 & 0.5 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (4)$$

For nominal or discrete values, symmetric similarity matrix is defined:

(i) Work. A similarity of 1 is considered for comparison with students; a person without a job vs. a student is given a value of 0.9; the value 0 is given to a person who has a job vs. a student. 0.9 is also given to a retired person vs. a person with a job, (Richter and Weber, 2016).

$$\mathrm{sim}_{\mathrm{work}}(x, y) = \begin{pmatrix} 1 & 0.9 & 0 & 0 \\ 0.9 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.9 \\ 0 & 0 & 0.9 & 1 \end{pmatrix} \quad (5)$$

(ii) Type of contract. In this case, an order of priority has been established: government official contracts are considered to be the most important type, followed by Permanent contract, Pension and Temporary. Therefore, a comparison between contract type 1 and 2 would give a value of 0.66; contract type 1 compared with 3 would give a value of 0.33 and contract type 1 compared with 4 would give a value of 0. The value of 1 is for values of the same category.

Formally:

$$\mathrm{sim}_n = 1 - \frac{|x-y|}{n-1} \tag{6}$$

where x and y are the order in the ranking and n the number of categories contained. When expressed as a matrix:

$$\mathrm{sim}_{\mathrm{contract}}(x,y) = \begin{pmatrix} 1 & 0.66 & 0.33 & 0 \\ 0.66 & 1 & 0.66 & 0.33 \\ 0.33 & 0.66 & 1 & 0.66 \\ 0 & 0.33 & 0.66 & 1 \end{pmatrix} \tag{7}$$

(iii) To work with attributes where a relevant value may not be defined, we extend their range of values to include a special "n/a" option (not available). When working with other values to measure the similarity between attributes the general rule will apply:

$$\mathrm{sim}(\mathrm{n/a}, x) = \mathrm{n/a} \tag{8}$$

Finally, when taking the weighted average of values that results in the total similarity, the individual components with value n/a are not taken into consideration, i.e.,

$$\mathrm{sim}(x,y) = \frac{\sum_{i\in I} \mathrm{sim}_i(x,y) w_i}{\sum_{i\in I} w_i}, \tag{9}$$

where I is the set of attributes with non-null values.

Thus each attribute will be weighted since none of the case attributes (k) are equally relevant. In addition, the problem of describing the importance of the case attributes is also denoted by a numerical value, as shown in Table 4.

To sum up this subsection, it could be said that the work proposed by Richter and Weber (2016) is used as a starting point to establish the similarity functions. The weights assigned to them arise from the authors' intuition and they can be further refined in a real scenario. The ideal situation would be to use a real dataset of existing recommendations, so that they can be formally adjusted.

The formal retrieval process is summarized in Alg. 1.

Algorithm 1 Retrieval of k most-similar users to c_0

```
function RETRIEVE(c_0, k)
    l ← List()                                  ▷ Similarity list
    for c ∈ CB do                               ▷ For each case in the case base
        Append(l, sim(c_0, c))                  ▷ Calculate the similarity (Eq. (9))
    end for
    l ← ArgSort(l)       ▷ Retrieve the indices with the values in decreasing order
    C ← List()                                  ▷ Output case list
    for i ∈ {1, …, k} do
        Append(C, CB[i])                        ▷ Fetch the i-th most similar case
    end for
    return C
end function
```

3.2.2. CEBRA Reuse stage. K-nearest neighbour classification performance

At this stage, the CBR makes a recommendation on the basis of the products most commonly purchased by the nearest neighbours in terms of their similarity function. In order to improve the recommendation capacity of the system, other recommendation techniques, such as collaborative filtering (CF), can be added additionally (Hernandez-Nieves et al., 2020).

3.2.3. CEBRA revise stage

As explained above, the expert will be provided with the interests that the users have shared publicly on their social networks, so that the adequacy of the proposal can be verified. The recommendation may be refined by checking similar profiles with similar preferences and interests that have purchased products that the subject of study has not contracted. In this example, as shown (Table 4), it is observed that the Query problem shows an overall similarity with problem 4 (*ini_state4*) of 0.862. We observe that the only difference is that the subject of problem 4 has no health insurance, so the recommendation given in our query problem (*ini_stateQ*) defined in (Table 3), would be to get it. Considering the interests extracted from the user's social network, it could be also established that the subject is a woman, she likes sports, and that she is a climbing enthusiast, therefore the recommendation could be geared towards an accident insurance in addition to hiring a health insurance.

4. Results: CEBRA implementation

CEBRA has been implemented as an API/REST, as shown in Fig. 4, for that Swagger UI (Varanasi and Belida, 2015). A synthetic dataset with 60,000 cases has been generated to test the functioning of CEBRA and simulation tools have been used for the case base construction. CEBRA is an application developed for banking entities to recommend products effectively and increase their sales. The source code is available in a public GitHub repository (Hernandez-Nieves, 2020). The objective was to build a case base that contains attributes that are actually used by banks, the part that really belongs to a bank is the dictionary from which we extracted the attributes, so no real data has been used to build the case base. It should be considered that the focus of the article is on architecture and design.

The Api allows the retrieval of the users that are most similar to the one to whom the recommendation is to be made. Therefore, this Api/retrieve endpoint corresponds to the Retrieve stage. Additionally, the api/recommend endpoint provides a recommendation to offer to the user. This recommendation could be completed with a system external to CEBRA, as it is shown in the Reuse stage in Fig. 2. The Review stage would be carried out externally to the system, where a team of experts would use all the information about the recovered cases, in addition to the measure of the similarities with those cases on which the recommendation has been based. Finally, when a positive recommendation has been made, the cases in CEBRA can be updated with the endpoint `/api/cases/{case_id}`, which would correspond to the Retain stage. The API is completed with the `/api/cases` endpoint, which allows to check the case base globally, and `/api/`, which allows to check the CEBRA configuration, including aspects such as the similarity measures incorporated in it.

When developing software, it is necessary to define requirements and verify them. User requirements describe requirements in a way that can be understood by users, usually defined using natural language, tables and diagrams. Requirements can have several origins, such as the domain of the problem (domain requirements). Domain requirements are user requirements that describe the characteristics and needs of the domain (common to all organisations in that sector). The problems that can arise with this type of requirement are mainly comprehensibility and misunderstandings. Comprehensibility because the requirements use the language and vocabulary usual in the domain of the application and are not correctly understood by the software engineers who are going to develop it and of course because on many occasions domain requirements are ignored as they are perfectly known by the experts in the area. The CBR CEBRA includes several domain requirements:

1. The user (the banking institution) must take into account the General Data Protection Regulation (EU 2016/279) on the protection of individuals with regard to the processing of personal data and to the free movement of such data (this text includes the corrigendum published in the DOUE of 23 May 2018).

Table 4
Weighing and importance of case attributes within each defined problem.

k	*ini_state*1	*ini_state*2	*ini_state*3	*ini_state*4	*ini_state*5	Importance
Age	0,72	0,15	0,77	0,99	0	3
Gender	0	0	0	1	0	3
Civil Status	1	1	0	1	0,5	3
City	1	1	1	1	1	4
Children	0,5	0,5	0	0,5	0,25	2
Work	1	0,9	1	1	0,9	7
Type of contract	0	0,66	1	0,33	0,66	7
No. of houses owned	1	1	0	0	1	4
No. of cars owned	0	0	1	0	1	4
No. other types vehicles owned	1	1	0	1	1	4
Current account	1	1	1	1	1	1
Payroll accounting	1	1	1	1	1	1
Has there ever been a default?	1	0	0	1	0	3
Regular defaults	1	1	0	1	1	5
Casual defaults	1	0	0	1	0	4
Positive end-of-month balance (usual, medium, low)	1	0	0,5	1	1	8
Saving account	1	0	1	1	0	3
Debit card	1	1	1	1	1	2
Credit card	1	0	0	1	1	2
Applied for a mortgage	0	0	0	1	0	9
Mortgage granted	n/a	n/a	n/a	1	n/a	3
Early mortgage payback	n/a	n/a	n/a	1	n/a	3
First time doing early mortgage payback	n/a	n/a	n/a	n/a	n/a	3
First time applying for a mortgage	n/a	n/a	n/a	1	n/a	3
Applied for a loan	0	1	0	1	0	9
Loan granted	n/a	n/a	n/a	n/a	n/a	3
First time applying for a Loan	n/a	n/a	n/a	n/a	n/a	3
Applied for a deposit	1	1	1	1	0	9
First time applying for a deposit	n/a	n/a	n/a	n/a	n/a	3
Applied for an investment fund	1	0	1	1	0	9
First time applying for an investment fund	n/a	n/a	n/a	n/a	n/a	3
Applied for a home insurance	1	0	0	1	0	9
First time applying for a home insurance	1	0	0	1	0	3
Applied for an accident insurance	1	1	1	1	1	9
First time applying for an accident insurance	n/a	n/a	n/a	n/a	n/a	3
Applied for a life insurance	1	0	0	1	0	9
First time applying for a life insurance	1	n/a	n/a	1	n/a	3
Applied for a health insurance	1	1	1	0	1	9
First time applying for a health insurance	n/a	n/a	n/a	n/a	n/a	3
Applied for a car/motorcycle insurance	1	1	0	1	1	9
First time applying for a car/motorcycle insurance	n/a	n/a	n/a	n/a	n/a	3
Overall similarity [Eq. (9)]	0.784	0.528	0.463	0.862	0.486	

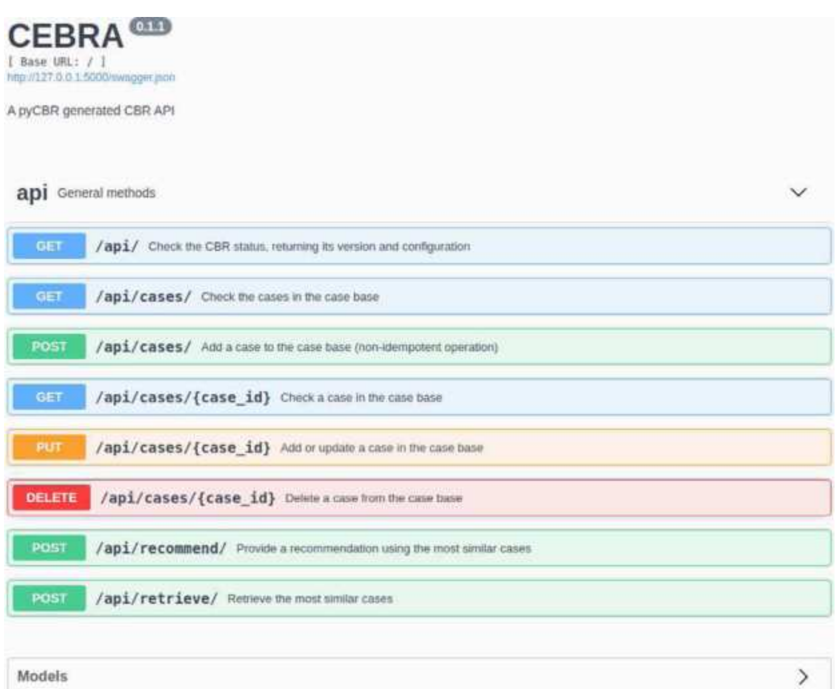

Fig. 4. CEBRA as an API/REST, displayed using Swagger UI.

2. Regulation (EU) 2018/1725 laying down the rules applicable to the processing of personal data by the institutions, bodies, offices and agencies of the Union should be taken into account.
3. The regulations specific to each country outside the European Union must also be taken into account if appropriate.

5. Conclusions and future work

CEBRA, a CBR case-base system has been described in this paper. CEBRA has been created to be incorporated into a Fog computing architecture that supports the commercial banking decision process using a combination of local and global decision models and local data. For the management and automation of the operation of CEBRA, the incorporation of a virtual organization of agents has been proposed. These Virtual Agent Organizations use their communication and co-ordination capabilities to share the results obtained by CEBRA. The main difference between CEBRA and other CBRs is the incorporation of several techniques within its life cycle. Its main contribution is that it creates a more complete user profile through the extraction of information from social networks. In the first phase of the cycle, in addition to collecting data, the interests of clients are extracted from their social networks by applying classifiers to the textual content. This information is later given to expert reviewers to complete the case. In the reuse phase, it is proposed to incorporate kNN to improve the recommendation capacity.

The article has introduced, designed and developed a framework as an Api/REST so that it can be implemented in a banking institution. If, however, a banking institution wants to implement it, ethical aspects and customer data privacy should be taken into account, as explained in Section 4. From the banking institution the database dictionary has been obtained, allowing for the consideration of the attributes to be selected for the case-base. The customer's data privacy rights have not been compromised at any time.

Regarding the analysis of the activities on social networks, it must be noted that we assume that the data and tweets shared on Twitter, which would be the social network proposed for this purpose, are public unless the user makes their account private. Within the privacy policy of Twitter, the following points are made clear: *Most activity on Twitter is public, which includes your profile information, your time zone and language, the date your account was created and your Tweets, as well as certain information about your Tweets such as the date, time and the application and version of Twitter from which you Tweet. You can also choose to publish your location in your Tweets or on your Twitter profile. The lists you create, the people you follow and the Tweets you do like or Retweets are also public. If you want to retweet, respond or interact publicly with an ad on our services, that marketer may get information about you associated with the ad you interacted with, such as the characteristics of the audience the ad was intended for. The Periscope transmissions you create, click on or otherwise participate in, whether on Periscope or Twitter, are public, along with information about when you performed such actions. So are the hearts, comments, the number of hearts you've received, which accounts you're a Superfan of and whether you watched a live or repeat broadcast. Any hearts, comments or other content that you contribute to another account's broadcast will remain part of that broadcast for as long as you remain on Periscope. Information posted about you by others using our services may also be public […]*.

When making a recommendation, the API makes it possible to retrieve the users that are highly similar to the one for whom the recommendation is being made. This API makes it possible to give a recommendation to the user, to update the cases and to check the case base in a global way. In conclusion, the developed application is capable of recommending products and it is provided to the academic community with the material required for its use and adaptation. We also provide the dataset that we have generated, so that the research community may use it.

CEBRA has been designed to be hosted in a Fog Computing environment governed by virtual agents that share business intelligence. Future lines of research will focus on designing a mechanism for agents that will enable them to adjust their CBR similarity weights to their operation. Therefore, in future research we will create a virtual environment where agents can perform this task.

CRediT authorship contribution statement

Elena Hernández-Nieves: Conceptualization, Formal analysis, Investigation, Writing - original draft, Visualization, Funding acquisition. **Guillermo Hernández:** Software, Formal analysis, Investigation. **Ana B. Gil-González:** Investigation, Methodology, Writing - review & editing. **Sara Rodríguez-González:** Conceptualization, Methodology, Writing - review & editing, Supervision. **Juan M. Corchado:** Conceptualization, Methodology, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The research of Elena Hernández-Nieves is supported by the Ministry of Education of the Junta de Castilla y León and the European Social Fund through a grant from predoctoral recruitment of research personnel associated with the University of Salamanca research project "ROBIN: Robo-advisor intelligent".

References

Aamodt, A., Plaza, E., 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. 7 (1), 39–59.
Artikis, A., 2009. Dynamic protocols for open agent systems. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems-Vol. 1, pp. 97–104.
Carrascosa, C., Giret, A., Julian, V., Rebollo, M., Argente, E., Botti, V., 2009. Service oriented MAS: an open architecture. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems-Vol. 2, pp. 1291–1292.
Chamoso, P., Rivas, A., Rodríguez, S., Bajo, J., 2018. Relationship recommender system in a business and employment-oriented social network. Inform. Sci. 433, 204–220.
Chen, D., Burrell, P., 2001. Case-based reasoning system and artificial neural networks: A review. Neural Comput. Appl. 10 (3), 264–276.
Consortium, O., et al., 2017. OpenFog Reference Architecture for Fog Computing. Architecture Working Group.
Corchado, J.M., Laza, R., 2003. Constructing deliberative agents with case-based reasoning technology. Int. J. Intell. Syst. 18 (12), 1227–1241.
Corchado, J.M., Pavón, J., Corchado, E.S., Castillo, L.F., 2004. Development of CBR-bdi agents: a tourist guide application. In: European Conference on Case-Based Reasoning. Springer, pp. 547–559.
De Mantaras, R.L., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., T COX, M., Forbus, K., et al., 2005. Retrieval, reuse, revision and retention in case-based reasoning. Knowl. Eng. Rev. 20 (3), 215–240.
Echeverry, G.A.I., Mejía, M.H., Ossa, L.F.C., Morales, A., Méndez, N.D.D., 2012. Network management using multi-agents system. ADCAIJ: Adv. Dist. Comput. Artif. Intell. J. 1 (3), 49–54.
Garcia-Fornes, A., Hübner, J.F., Omicini, A., Rodriguez-Aguilar, J.A., Botti, V., 2011. Infrastructures and tools for multiagent systems for the new generation of distributed systems. Eng. Appl. Artif. Intell. 24 (7), 1095–1097.
Hernandez-Nieves, E., 2020. CEBRA. https://github.com/ElenaHN/CEBRA.
Hernandez-Nieves, E., Hernández, G., Gil-González, A.-B., Rodríguez-González, S., Corchado, J.M., 2020. Fog computing architecture for personalized recommendation of banking products. Expert Syst. Appl. 140, 112900.
Jarrar, Y.F., Neely, A., 2002. Cross-selling in the financial sector: customer profitability is key. J. Target. Meas. Anal. Mark. 10 (3), 282–296.
Jubair, M.A., Mostafa, S.A., Mustapha, A., Hafit, H., 2018. A survey of multi-agent systems and case-based reasoning integration. In: 2018 International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR). IEEE, pp. 1–6.
Kolodner, J.L., Leake, D., 1996. A tutorial introduction to case-based reasoning.
de Mantaras, R.L., 1999. Case-based reasoning. In: Advanced Course on Artificial Intelligence. Springer, pp. 127–145.
Melnychenko, S., Volosovych, S., Baraniuk, Y., 2020. Dominant ideas of financial technologies in digital banking. Balt. J. Econ. Stud. 6 (1), 92–99.

Naumov, M., Mudigere, D., Shi, H.-J.M., Huang, J., Sundaraman, N., Park, J., Wang, X., Gupta, U., Wu, C.-J., Azzolini, A.G., et al., 2019. Deep learning recommendation model for personalization and recommendation systems. ArXiv Preprint arXiv:1906.00091.

Nikzad-Khasmakhi, N., Balafar, M., Feizi-Derakhshi, M.R., 2019. The state-of-the-art in expert recommendation systems. Eng. Appl. Artif. Intell. 82, 126–147.

Oyenan, W.H., DeLoach, S.A., Singh, G., 2009. Exploiting reusable organizations to reduce complexity in multiagent system design. In: International Workshop on Agent-Oriented Software Engineering. Springer, pp. 3–17.

Richter, M.M., Weber, R.O., 2016. Case-Based Reasoning. Springer.

Riesbeck, C.K., Schank, R.C., 2013. Inside Case-Based Reasoning. Psychology Press.

Rodriguez, S., Julián, V., Bajo, J., Carrascosa, C., Botti, V., Corchado, J.M., 2011. Agent-based virtual organization architecture. Eng. Appl. Artif. Intell. 24 (5), 895–910.

Roth-Berghofer, T., Recio Garcia, J.A., Sauer, C., Bach, K., Althoff, K.-D., Diaz-Agudo, B., Gonzales Calero, P.A., 2012. Building case-based reasoning applications with mycbr and colibri studio.

Skjold, K., Øynes, M.S., 2017. Case-Based Reasoning and Computational Creativity in a Recipe Recommender System. NTNU.

Sridevi, M., Rao, R.R., Rao, M.V., 2016. A survey on recommender system. Int. J. Comput. Sci. Inf. Secur. 14 (5), 265.

Varanasi, B., Belida, S., 2015. Documenting rest services. In: Spring REST. Springer, pp. 91–104.