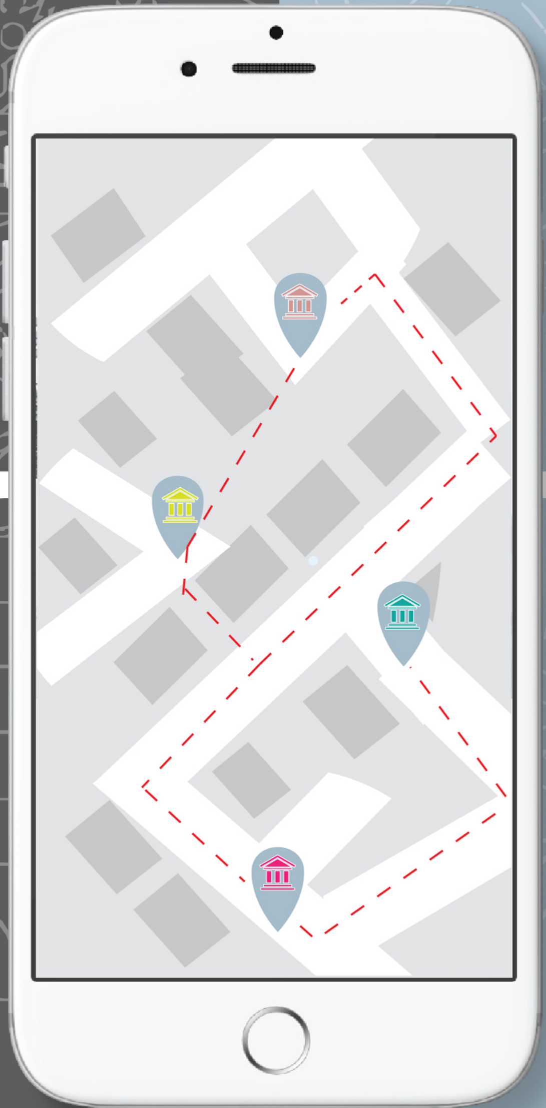

Pubblicazione edita a cura del Dipartimento di Ingegneria Industriale

ISBN 88-7897-122-7

Methods and Systems for Context Awareness in Complex Scenarios: the Case of Cultural Heritage Sites

MARCO LOMBARDI

UNIVERSITÀ DEGLI STUDI DI SALERNO

University of Salerno
Department of Industrial Engineering

METHODS AND SYSTEMS FOR CONTEXT AWARENESS IN COMPLEX SCENARIOS: THE CASE OF CULTURAL HERITAGE SITES

Marco Lombardi

Methods and Systems for Context Awareness in Complex Scenarios: the Case of Cultural Heritage Sites

Marco Lombardi

UNIVERSITY OF SALERNO

DEPARTMENT OF INDUSTRIAL ENGINEERING

Ph.D. Course in Industrial Engineering
Curriculum in Electronic Engineering - XXXIII Cycle

Methods and Systems for Context Awareness in Complex Scenarios: the Case of Cultural Heritage Sites

Supervisor
Prof. Francesco Colace

Ph.D. student
Marco Lombardi

Scientific Referees
Prof. Arcangelo Castiglione
Prof. Vincenzo Moscato
Ing. Alfredo Troiano

Ph.D. Course Coordinator
Prof. Francesco Donsì

“No one can whistle a symphony.

It takes an orchestra to play it.”

– H. E. Luccock

Publications List

Journal Articles

M. Lombardi, F. Pascale, D. Santaniello: "Internet of Things: A General Overview between Architectures, Protocols and Applications". *INFORMATION, MDPI, 2021, Vol. 12, No. 2, Pag. 87 (1-20).* DOI: 10.3390/info12020087

F. Colace, M. De Santo, M. Lombardi, R. Mosca, D. Santaniello: "A Multilayer Approach for Recommending Contextual Learning Paths". *Journal of Internet Services and Information Security (JISIS), 2020, Vol. 10, No. 2, Pag. 91-102.* DOI: 10.22667/JISIS.2020.05.31.091

A. Castiglione, F. Palmieri, F. Colace, M. Lombardi, D. Santaniello, G. D'Aniello: "Securing the Internet of Vehicles through Lightweight Block Ciphers". *Pattern Recognition Letters, 2020, Vol. 135, Pag. 264-270.* DOI: 10.1016/j.patrec.2020.04.038

M. Casillo, F. Clarizia, G. D'Aniello, M. De Santo, M. Lombardi, D. Santaniello: "CHAT-Bot: a Cultural Heritage Aware Teller-Bot for supporting touristic experiences". *Pattern Recognition Letters, 2020, Vol. 131, Pag. 234-243.* DOI: 10.1016/j.patrec.2020.01.003

M. Casillo, F. Clarizia, F. Colace, M. Lombardi, F. Pascale, D. Santaniello: "An approach for recommending contextualized services in e-tourism". *INFORMATION, MDPI, 2019, Vol. 10, Pag. 180 (1-17).* DOI: 10.3390/INFO10050180

F. Clarizia, F. Colace, M. De Santo, M. Lombardi, F. Pascale, D. Santaniello, D.; Tuker, A.: "A multilevel graph approach for rainfall forecasting: A preliminary study case on London area". *Concurrency and Computation –*

Practice and Experience (CCPE), 2019 Vol. 32, No. 8. DOI: 10.1002/cpe.5289

M. Casillo, F. Colace, M. De Santo, S. Lemma, M. Lombardi: "CAT: a Context Aware Teller for supporting tourist experiences". *International Journal of Computational Science and Engineering (IJCSE), 2019, Vol. 20, No. 1, Pag. 69-87.* DOI: 10.1504/IJCSE.2017.10013620

F. Colace, M. De Santo, S. Lemma, M. Lombardi, F. Pascale, A. Pietrosanto: "Chatbot for E-Learning: A Case of Study". *International Journal of Mechanical Engineering and Robotics Research (IJMERR), 2018, Vol. 7, No. 5.* DOI: 10.18178/ijmerr.7.5.528-533

F. Colace, M. De Santo, S. Lemma, M. Lombardi, F. Pascale: "SmartApp Salemo: an Adaptive Context Aware App for supporting e tourists". *Research Briefs on Information & Communication Technology Evolution (ReBICTE), 2017, Vol. 3, No. 2.* DOI: 10.22667/reBiCTE.2017.05.15.002

Book Chapters

M. Casillo, D. Conte, M. Lombardi, D. Santaniello, C. Valentino: "Recommender System for Digital Storytelling: A Novel Approach to Enhance Cultural Heritage". *(2021) In: Del Bimbo A. et al. (eds) 25th International Conference on Pattern Recognition (ICPR2020), 2021 – 2nd International Workshop on Pattern Recognition for Cultural Heritage (PatReCH2020). Pattern Recognition. ICPR International Workshops and Challenges. Lecture Notes in Computer Science. Springer, Cham, vol. 12667, p. 304-317.* DOI: 10.1007/978-3-030-68787-8_22

F. Clarizia, M. De Santo, M. Lombardi, R. Mosca, D. Santaniello: "A Contextual Approach for Coastal Tourism and Cultural Heritage Enhancing". *(2021) In: Del Bimbo A. et al. (eds) 25th International Conference on Pattern Recognition (ICPR2020), 2021 – 2nd International Workshop on Pattern Recognition for Cultural Heritage (PatReCH2020). Pattern Recognition. ICPR International Workshops and Challenges. Lecture Notes in Computer Science. Springer, Cham, vol. 12667, p. 318-325.* DOI: 10.1007/978-3-030-68787-8_23

M. Casillo, F. Colace, M. De Santo, M. Lombardi, D. Santaniello: "Salerno Telling: An Approach for Cultural Heritage Enhancing". *(2021) In: Visvizi, A., Lytras, M.D., Aljohani, N.R. (eds) Research and Innovation Forum*

(RIIFORUM), 2020. Springer Proceedings in Complexity, p. 103-112. DOI: 10.1007/978-3-030-62066-0_9

F. Clarizia, F. Colace, M. De Santo, M. Lombardi, D. Santaniello: "A Recommender System for Enhancing Coastal Tourism". *(2021) In: Visvizi, A., Lytras, M.D., Aljohani, N.R. (eds) Research and Innovation Forum (RIIFORUM), 2020. Springer Proceedings in Complexity, p. 113-122.* DOI: 10.1007/978-3-030-62066-0_10

M. Casillo, F. Colace, M. De Santo, M. Lombardi, R. Mosca, D. Santaniello: "A Chatbot for Training Employees in Industry 4.0". *(2021) In: Visvizi, A., Lytras, M.D., Aljohani, N.R. (eds) Research and Innovation Forum (RIIFORUM), 2020. Springer Proceedings in Complexity, p. 397-409.* DOI: 10.1007/978-3-030-62066-0_30

F. Clarizia, M. De Santo, M. Lombardi, D. Santaniello: "E-Learning and Industry 4.0: A Chatbot for Training Employees". *(2021) In: Yang XS., Sherratt S., Dey N., Joshi A. (eds) 5th International Congress on Information and Communication Technology (ICICT), 2020. Advances in Intelligent Systems and Computing. Springer, Singapore, vol. 1184, p. 445-453.* DOI: 10.1007/978-981-15-5859-7_44

F. Colace, M. Khan, M. Lombardi, D. Santaniello: "A Multigraph Approach for Supporting Computer Network Monitoring Systems". *(2021) In: Yang XS., Sherratt S., Dey N., Joshi A. (eds) 5th International Congress on Information and Communication Technology (ICICT), 2020. Advances in Intelligent Systems and Computing. Springer, Singapore, vol. 1184, p. 470-477.* DOI: 10.1007/978-981-15-5859-7_46

M. Casillo, F. Colace, D. Conte, M. De Santo, M. Lombardi, S. Mottola, D. Santaniello: "A Multi-feature Bayesian Approach for Fake News Detection". *(2020) In: Chellappan S., Choo KK.R., Phan N. (eds) 9th International Conference on Computational Data and Social Networks (CSoNet), 2020. Lecture Notes in Computer Science. Springer, Cham, vol. 12575, p. 333-344.* DOI: 10.1007/978-3-030-66046-8_27

F. Colace, M. De Santo, M. Lombardi, F. Pascale, D. Santaniello, A. Tucker: "A Multilevel Graph Approach for Predicting Bicycle Usage in London Area". *(2020) In: Yang XS., Sherratt S., Dey N., Joshi A. (eds) 4th International Congress on Information and Communication Technology (ICICT), 2019. Advances in Intelligent Systems and Computing. Springer, Singapore, vol. 1027, p. 353-362.* DOI: 10.1007/978-981-32-9343-4_28

F. Clarizia, F. Colace, M. Lombardi, F. Pascale, D. Santaniello: "Chatbot: An Education Support System for Student". *In: A. Castiglione, F. Palmieri, M. Ficco, F. Pop. (eds) 10th International Symposium on Cyberspace Safety and Security (CSS), 2018. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Verlag, vol. 11161, p. 291-302.* DOI: 10.1007/978-3-030-01689-0_23

F. Clarizia, F. Colace, M. Lombardi, F. Pascale, D. Santaniello, Domenico: "A Multilevel Graph Approach for Road Accidents Data Interpretation". *In: A. Castiglione, F. Palmieri, M. Ficco, F. Pop. (eds) 10th International Symposium on Cyberspace Safety and Security (CSS), 2018. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Verlag, vol. 11161, p. 303-316.* DOI: 10.1007/978-3-030-01689-0_24

F. Clarizia, S. Lemma, M. Lombardi, F. Pascale: "A Mobile Context-Aware Information System to Support Tourism Events". *In: M. Au, A. Castiglione, KK. Choo, F. Palmieri, KC Li. (eds) 12th International Conference on Green, Pervasive and Cloud Computing (GPC), 2017. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Cham, vol. 10232, p. 553-566.* DOI: 10.1007/978-3-319-57186-7_40

F. Clarizia, S. Lemma, M. Lombardi, F. Pascale: "An Ontological Digital Storytelling to Enrich Tourist Destinations and Attractions with a Mobile Tailored Story". *In: M. Au, A. Castiglione, KK. Choo, F. Palmieri, KC Li. (eds) 12th International Conference on Green, Pervasive and Cloud Computing (GPC), 2017. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Cham, vol. 10232, p. 567-581.* DOI: 10.1007/978-3-319-57186-7_41

Conference Proceedings

M. Casillo, F. Colace, L. Fabbri, M. Lombardi, A. Romano, D. Santaniello: "Chatbot in Industry 4.0: An Approach for Training New Employees". *2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), 2020, 371-376.* DOI: 10.1109/TALE48869.2020.9368339

M. De Santo, L. Fabbri, R. Mosca, M. Lombardi, A. Romano, D. Santaniello: "A Multilevel Approach to Recommend Working Paths in Industry 4.0". *2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), 2020, 754-757.* DOI: 10.1109/TALE48869.2020.9368327

M. Lombardi, F. Pascale, D. Santaniello: "EIDS: Embedded Intrusion Detection System using Machine Learning to detect attack over the CAN-BUS". *30th European Safety and Reliability Conference (ESREL), 2020, 2028-2035.* DOI: 10.3850/978-981-14-8593-0_5090-cd

F. Clarizia, F. Colace, M. Lombardi, D. Santaniello: "A Chatbot for supporting users in Cultural Heritage contexts". *26th International DMS Conference on Visualization and Visual Languages (DMSVIVA), 2020, 22-27.* DOI: 10.18293/DMSVIVA20-009

A. Castiglione, F. Palmieri, F. Colace, M. Lombardi, D. Santaniello: "On the impact of lightweight ciphers in automotive networks". *26th International DMS Conference on Visualization and Visual Languages (DMSVIVA), 2020, 87-92.* DOI: 10.18293/DMSVIVA20-010

F. Amato, M. Casillo, F. Colace, M. De Santo, M. Lombardi, D. Santaniello: "CHAT: a Cultural Heritage Adaptive Tutor". *2019 IEEE International Conference on Engineering, Technology and Education (TALE), 2019, 1-5.* DOI: 10.1109/TALE48000.2019.9225962

F. Clarizia, F. Colace, M. De Santo, M. Lombardi, F. Pascale, D. Santaniello: "A Context-Aware Chatbot for Tourist Destinations". *15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2019, 348-354.* DOI: 10.1109/SITIS.2019.00063

A. Castiglione, F. Palmieri, F. Colace, M. Lombardi, D. Santaniello: "Lightweight Ciphers in Automotive Networks: A Preliminary Approach". *4th International Conference on System Reliability and Safety (ICSRS), 2019, 142-147.* DOI: 10.1109/ICSRS48664.2019.8987693

F. Colace, M. De Santo, M. Lombardi, D. Santaniello: "CHARS: a Cultural Heritage Adaptive Recommender System". *1st ACM International Workshop on Technology Enablers and Innovative Applications for Smart Cities and Communities (TESCA), 2019, 58-61.* DOI: 10.1145/3364544.3364830

F. Clarizia, F. Colace, M. Lombardi, F. Pascale, D. Santaniello: "Sentiment Analysis in Social Networks: A Methodology Based on the Latent Dirichlet Allocation Approach". *2019 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (EUSFLAT), 2019, 241-248.* DOI: 10.2991/eusflat-19.2019.36

M. Casillo, F. Clarizia, F. Colace, M. De Santo, M. Lombardi, F. Pascale: "A Latent Dirichlet Allocation Approach using Mixed Graph of Terms for Sentiment Analysis". *52nd Hawaii International Conference on System Sciences (HICSS), 2019, 2226-2234.* DOI: 10.24251/HICSS.2019.270

F. Colace, M. De Santo, M. Lombardi, F. Mercorio, M. Mezzanzanica, F. Pascale: "Towards Labour Market Intelligence through Topic Modelling". *52nd Hawaii International Conference on System Sciences (HICSS), 2019, 5256-5265.* DOI: 10.24251/HICSS.2019.632

M. Lombardi, F. Pascale, D. Santaniello: "An application for Cultural Heritage using a Chatbot". *2nd International Conference on Computer Applications & Information Security (ICCAIS), 2019, 1-5.* DOI: 10.1109/CAIS.2019.8769525

M. Lombardi, F. Pascale, D. Santaniello: "A Double-layer Approach for Historical Documents Archiving". *4th International Conference on Metrology for Archaeology and Cultural Heritage (MetroArchaeo), 2018, 137-140.* DOI: 10.1109/MetroArchaeo43810.2018.13607

F. Colace, M. Lombardi, F. Pascale, D. Santaniello: "A Multilevel Graph Representation for Big Data Interpretation in Real Scenarios". *3rd International Conference on System Reliability and Safety (ICSRS), 2018, 40-47.* DOI: 10.1109/ICSRS.2018.8688834

F. Clarizia, F. Colace, M. De Santo, M. Lombardi, F. Pascale: "A Sentiment Analysis Approach for Evaluation of Events in Field of Cultural Heritage". *5th International Conference on Social Networks Analysis, Management and Security (SNAMS), 2018, 120-127.* DOI: 10.1109/SNAMS.2018.8554959

F. Colace, M. Lombardi, F. Pascale, D. Santaniello, A. Tucker, P. Villani: "MuG: A Multilevel Graph Representation for Big Data Interpretation". *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on*

Smart City; IEEE 4th Intl. Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2018, 1408-1413. DOI: 10.1109/HPCC/SmartCity/DSS.2018.00233

F. Clarizia, F. Colace, M. Lombardi, F. Pascale: "A Sentiment Analysis Approach for supporting Blended Learning Process". *24th International DMS Conference on Visualization and Visual Languages (DMSVIVA), 2018, 8-14.* DOI: 10.18293/DMSVIVA2018-001

F. Colace, M. Lombardi, F. Pascale, D. Santaniello: "A multi-level approach for forecasting critical events in Smart Cities". *24th International DMS Conference on Visualization and Visual Languages (DMSVIVA), 2018, 31-35.* DOI: 10.18293/DMSVIVA2018-002

F. Clarizia, F. Colace, M. Lombardi, F. Pascale: "A Context Aware Recommender System for Digital Storytelling". 2018 *IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), 2018, 542-549.* DOI: 10.1109/AINA.2018.00085

F. Clarizia, F. Colace, M. De Santo, M. Lombardi, F. Pascale, A. Pietrosanto: "E-Learning and Sentiment Analysis: A Case Study". *6th International Conference on Information and Education Technology (ICIET), 2018, 111-118.* DOI: 10.1145/3178158.3178181

M. Casillo, F. Colace, S. Lemma, M. Lombardi, F. Pascale: "Context-aware computing for improving the touristic experience: A pervasive app for the Amalfi coast". *2017 IEEE International Workshop on Measurements and Networking (M&N), 2017.* DOI: 10.1109/IWMN.2017.8078373

F. Colace, M. De Santo, S. Lemma, M. Lombardi, F. Pascale: "BotWheels: a Petri Net based Chatbot for Recommending Tires". *3rd edition of Special Session on Knowledge Discovery meets Information Systems: Applications of Big Data Analytics and BI - methodologies, techniques and tools (KomIS), within the 6th International Conference on Data Science, Technology and Applications (DATA), 2017, 350-358.* DOI: 10.5220/0006491903500358

F. Colace, S. Lemma, M. Lombardi: "Context Awareness for e-Tourism: an Adaptive Mobile Application". *23rd International DMS Conference on Visual Languages and Sentient Systems (DMSVLSS), 2017, 67-74.* DOI: 10.18293/DMS2017-010

M. Casillo, F. Colace, S. Lemma, M. Lombardi, F. Pascale: "A Tailor made System for providing Personalized Services". *29th International Conference on Software Engineering & Knowledge Engineering (SEKE), 2017, 495-500.* DOI: 10.18293/SEKE2017-151

F. Colace, S. Lemma, M. Lombardi, F. Pascale: "A Context Aware Approach for Promoting Tourism Events: The Case of Artist's Lights in Salerno". *19th International Conference on Enterprise Information Systems (ICEIS), 2017, 752-759.* DOI: 10.5220/0006370007520759

F. Colace, M. De Santo, S. Lemma, M. Lombardi: "An Adaptive App for tourist contents contextualization". *2nd International Conference on Internet of things and Cloud Computing (ICC), 2017, 80.* DOI: 10.1145/3018896.3025128

M. Casillo, F. Colace, M. De Santo, S. Lemma, M. Lombardi: "A Context-Aware Mobile Solution for Assisting Tourists in a Smart Environment". *50th Annual Hawaii International Conference on System Sciences (HICSS), 2017, 495-504.* DOI: 10.24251/HICSS.2017.061

M. Casillo, F. Colace, M. De Santo, S. Lemma, M. Lombardi: "A Context-Aware Approach for a Collaborative, Pervasive and Adaptive Digital Storytelling". *22nd International Conference on Distributed Multimedia Systems (DMS), 2016, 165-174.* DOI: 10.18293/DMS2016-033

M. Casillo, L. Cerullo, F. Colace, S. Lemma, M. Lombardi, A. Pietrosanto: "An Adaptive Context Aware App for the Tourism". *3rd Multidisciplinary International Social Networks Conference (MISNC), 2016, 26.* DOI: 10.1145/2955129.2955150

M. Casillo, F. Colace, M. De Santo, S. Lemma, M. Lombardi, A. Pietrosanto: "An Ontological Approach to Digital Storytelling". *3rd Multidisciplinary International Social Networks Conference (MISNC), 2016, 27.* DOI: 10.1145/2955129.2955147

G. Annunziata, F. Colace, M. De Santo, S. Lemma, M. Lombardi: "ApPoggiomarino: A Context Aware App for E-Citizenship". *18th International Conference on Enterprise Information Systems (ICEIS), 2016, 273-281.* DOI: 10.5220/0005825202730281

F. Colace, L. Greco, S. Lemma, M. Lombardi, F. Amato, V. Moscato, A. Picariello: "Contextual Aware Computing and Tourism: a Case Study".

11th International Conference on Signal Image Technology & Internet Based Systems (SITIS), 2015, 804-808. DOI: 10.1109/SITIS.2015.49

F. Colace, M. De Santo, S. Lemma, M. Lombardi, A. Rossi, A. Santoriello, A. Terribile, M. Vigorito: "How to Describe Cultural Heritage Resources in the Web 2.0 Era?". *11th International Conference on Signal Image Technology & Internet Based Systems (SITIS), 2015, 809-815.* DOI: 10.1109/SITIS.2015.50

S.K. Chang, D. Yung, F. Colace, L. Greco, S. Lemma, M. Lombardi: "An Adaptive Contextual Recommender System: a Slow Intelligence Perspective". *27th International Conference on Software Engineering and Knowledge Engineering (SEKE), 2015, 64-71.* DOI: 10.18293/SEKE2015-080

F. Colace, M. De Santo, L. Greco, S. Lemma, M. Lombardi, V. Moscato, A. Picariello: "A Context-Aware Framework for Cultural Heritage Applications". *10th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2014, 469-476.* DOI: 10.1109/SITIS.2014.14

Summary

Figures Index

Tables Index

Abstract

Context-Aware Computing describes the development of technologies and applications that can detect data from the surrounding environment and react accordingly with specific actions, reducing and simplifying the human-machine interaction process. The latter automatically offer a range of services to help the user during daily professional or private life by managing the available resources. Therefore, Context Awareness (CA) should be intended as a set of technical features able to provide added value to services in different application segments.

Context changes result in a transformation of the user experience. For this reason, Context-Aware Computing has played an essential role in addressing this challenge in previous paradigms, such as Mobile and Pervasive Computing, and is playing a crucial role in the new Internet of Things (IoT) paradigm.

Over the last years, e-Tourism and, in particular, Cultural Heritage have provided two main domains for this type of research. Indeed, thanks to new technologies, a tourist can access large amounts of contents and services before, during and after visiting experience, with different purposes and requirements in each phase. In this scenario, the need arises for Recommendation Systems (RS) that consider users' personal preferences and all the contextual aspects to recommend the right services and contents at a specific time.

The research activity concerned the study of Context-Aware Recommender Systems (CARS), focusing on the modelling and managing all the possible Contexts in an application domain. In particular, the problem of tailored data modelling has arisen, as this represents an enabler for new information systems: Mobile Systems, Big Data Systems, P2P Systems and, in general, the Semantic Web.

In that regard, a system architecture for the fruition of e-Tourism contents and services was designed to enhance the Cultural Heritage by responding in a unique way to the Context and users' needs. It will be capable of supporting not only visiting users but also public institutions and sector operators through the automatic as well as the dynamic definition and recommendation of core

and ancillary services for tourism promotion: from the search for a destination up to the use of Cultural Heritage-related content and the commentary of the visitor experience, including tourism promotion services, booking, e-ticketing, e-commerce, social networking.

To recommend contextual contents and services, the innovative characteristics of the proposed approach mainly concern the information to be made available to end-users, suggesting three main points of view:

1. Data Management and Inferential Engines

In such a scenario, data represent the key to build up and enable services and actions to take: the goal is to implement a Knowledge Base (KB) to collect, elaborate, and manage information in real-time. In this respect, Knowledge Organization Systems (KOS) refer to well-known schemes such as taxonomies, thesauri and other types of vocabularies that, together with ontologies, constitute valuable tools to shape the reality of interest into concepts and relations between concepts.

The system, thought to be continuously functioning, collects data from various sources without interruption and immediately processes them, intending to activate precise actions, depending on the users and the events. The latter, detected and analysed, will have to be translated into facts associated with specific semantic values: it is necessary to use inferential engines capable of drawing some conclusions by applying particular rules to reported facts. In this regard, many approaches are based on the so-called Bayesian Networks: powerful conceptual, mathematic and application tools allowing the management of complex problems with a significant number of variables interlinked by both probabilistic and deterministic relations. Such networks also make it possible to update the probabilities of all the variables involved whenever new information is collected on some of them, using Bayes' theorem.

2. Context Representation

The goal is primarily to deliver to different categories of users, in each moment, information that is useful in a given Context. In practice, the objective would be setting up an architecture characterised by a high degree of Context-Awareness. Real-time understanding of the Context where users are located, via a representation by means of graphs, allows indeed to provide them with a wide array of "tailored" services and hints regarding the decisions to make, managing in the best possible way both the time and resources they have and showing them what is around, ultimately meeting their needs. More in detail, the Context's representation can be implemented through formal models of representation, such as the Context Dimension Tree (CDT).

3. Recommender Systems

A Context-Aware System's ability to reduce information noise takes on considerable importance together with the possibility of the system itself generating an ordered list of personalised suggestions in each Context through a recommendation engine. Recommender Systems are applied in different sectors but have one goal: to help people make choices based on an analysis of users and items in terms of main features. In other words, the purpose is to predict the consideration that an individual may have about an object that he has not yet evaluated.

Ultimately, the goal is to identify a framework, mainly based on a powerful contextual recommendation engine, to be a highly flexible inferential and decision-making tool. This framework does not only allow to manage of complex problems, featuring a great variety of variables inter-linked through both logical-deterministic and probabilistic relationships, but it also provides an adequate representation of the phenomenon at stake. In fact, it simplifies the problem description as well as the summary easier, enhancing the degree of its comprehension and allowing to identify the key variables. In addition, modularity allows for easy integration of new functionalities that can be developed and tested separately, such as a process capable of presenting information in learning environments according to Digital Storytelling techniques.

Based on the proposed architecture, an application prototype was developed to support the user in the construction of a personalised and contextualised tourist route related to some of the most important cultural sites in Campania (a region in Southern Italy): a hybrid mobile application designed and implemented together with a server-side component.

The experimental results show the ability of the system to be effective. Future activities include improving the developed prototype, including a chatbot, and an experimental campaign involving a more significant number of users.

Chapter 1
Context Awareness

The importance of Context in Information Technology has increased in recent decades as computer systems have become more pervasive in everyday life. Context Awareness, meaning the idea that such systems can detect and react to a user's situation, constitutes a popular research topic.

While the IT community has initially considered the Context as a matter of user position (Abowd *et al.*, 1999), in recent years, this concept has been considered as part of the whole process in which users are involved (Coutaz *et al.*, 2005). Sophisticated and general Context models have been put forward to support context-aware applications, which use them, for example, (1) to adapt the interfaces (De Virgilio *et al.*, 2006), (2) to tailor a data series relevant to the application (Bolchini *et al.*, 2006), (3) to increase the accuracy of information collection (Shen *et al.*, 2005), (4) to discover services (Raverdy *et al.*, 2006), (5) to make the user's interaction implicit (Petrelli *et al.*, 2001), (6) to create intelligent environments (Dey *et al.*, 2006).

Consider the example of automated support for museum visitors, equipped with a mobile device that reacts to Context changes: (a) by adapting the user interface according to the different abilities of the visitor; (b) by providing different informative contents based on the different profiles and interests of visitors (such as students, journalists and archaeologists) and on the room where they currently are; (c) by predicting, based on the previous choices made by the visitor, the information he/she will be looking for at a later time; (d) by providing appropriate services, for example, to purchase a ticket for a temporary exhibition or to reserve a seat for the next exhibition on their favourite author's life; (e) by drawing position-related information from sensors detecting the user's environment; (f) by providing active functions within the various areas of the museum, indicating to visitors suggestions and inputs on what is happening in each specific environment.

1.1 Background

The contextualisation of environmental variables within an interactive scenario has always been a topic of great interest: people have always tried to conceive the nature around them as an integral part of their action. In fact, when people talk to each other, this information is implicitly transmitted through the verbal description of the Context in which they find themselves.

Unfortunately, this case is not the case when people interact with electronic devices such as computers or mobile devices in general. When people interact with an electronic device, they cannot always provide all possible information and, therefore, there is an impoverishment of the interaction. For this reason, it is essential to be able to collect as much information as possible out of Context to have an entire interaction with the device used. Hence the need to have a method to analyse, understand and process all the necessary information to contextualise humans and machines' interaction. The main problem is to treat the various raw data from the sensors and make it useful as Context data.

1.1.1 Definition of Context

The concept of Context does not have a rigorous definition, and, over the years, many different definitions have been provided about its meaning. This concept indeed plays an essential role in many different disciplines, such as psychology, linguistics, and computer science. In each of these, it can take on a different meaning, more suited to its application.

- In Computer Science, the first to introduce the concept of "context-aware" were Schilit and Theimer (Theimer and Schilit, 1994), within their work on Distributed Mobile Computing. Their interpretation applied to the problem of "location awareness" in an office environment. The definition of Context included the places and identities of people, as well as the state of the objects inside the environment.

- The definition of Pascoe et al. instead included environmental characteristics, such as the description of the current weather conditions (Pascoe *et al.*, 1999). This definition was then remarked upon by Schmidt et al., who presented the concept of Context to include environmental conditions and information on devices, users, and their activities (Schmidt *et al.*, 1999).

- Chen and Kotz then emphasized the importance of "time context", such as time of a day, week, month and season of the year (Chen and Kotz, 2000).

- Lastly, Dey and Abowd defined the Context as follows: «Context is any information that can be used to characterize the situation of an entity. An entity is a person or an object that is considered relevant to the interaction between a user and an application, including the user and applications themselves» (Dey, 2001).

Among the various definitions of Context, this latter is considered the de facto standard in Computer Science. It clearly shows that if a piece of information can describe and feature the situation in which one of the interaction participants is located, it can be considered part of the Context.

However, given its marked abstraction, Dey and Abowd's definition is challenging to develop concretely without classification or a modelling system.

1.1.2 Context Classification

Similar to the definition of Context, there is no standard classification system.

- Chen and Kotz have classified Context into two abstraction levels (Chen and Kotz, 2000).

  At a higher level, the first was limited to identifying the importance of the surrounding environment. This classification was divided into:

  - *Active Context*: the set of characteristics of the surrounding environment that directly influences an application behaviour.

  - *Passive Context*: the set of characteristics of the peripheral environment that are still relevant for the application.

  At a lower level, the second looked at the Context as a four-dimensional space composed of:

  - *Computing Context*: concerns the technical aspects related to resources and computational capacity. This category has a dual purpose: 1) to express all the heterogeneities usually present in mobile environments, such as the different abilities and connectivity of the devices; 2) to consider the various resources that a mobile device meets while it moves.

- *Physical Context*: it gathers all the aspects representing the real world and accessible through sensors located nearby. A relevant example is the user's location (more specifically, the device's). Physical Context can also be lighting and noise levels, traffic conditions, and temperatures.
- *Temporal Context*: it captures the time dimension of any system activity such as the time, the day of the week, the month and the season by establishing an actual "temporal location". The variations of state belonging to Temporal Context are subdivided into two sub-categories, i.e. random and periodic.
- *User Context*: it covers every high-level aspect related to the social dimension of the user, such as the user's profile, the people nearby and the current social situation. As a matter of fact, Context-Aware Systems are distributed mobile systems envisaging the presence of multiple end-users. Therefore, each node has a particular Context, deriving from the objective view of the surrounding environment, and a social Context, deriving from the awareness of being an actor in the whole system.

- Instead, Dey e Abowd divided the Context into the following key categories (Abowd *et al.*, 1999): position, identity, time and activity. More recently, as shown in Figure 1.1, Zimmermann et al. extended these categories, holding that a social aspect should also define the Context through interactions between the entities: relations (Zimmermann *et al.*, 2007).

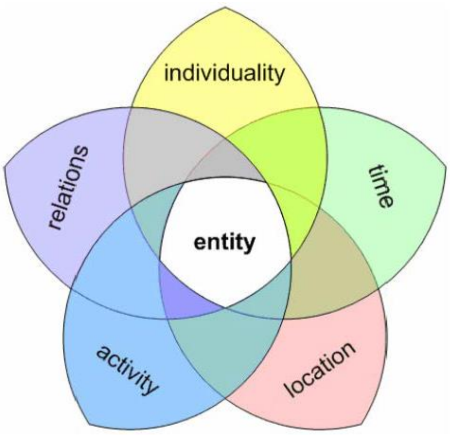

Figure 1.1 *Context categories according to Zimmermann*

Finally, a modern approach to Context definition and its subdivision into main aspects follows an informal information gathering method mainly used by journalists to report facts: 5W1H method (Who, What, Where, When, Why, How) (Dey *et al.*, 2006; Colace, Lemma, *et al.*, 2017). For example:

- *Who are we trying to identify? Who is using our application?* The "who" aspect refers to the user's identity, one of the main categories of Context proposed by Dey and Abowd. Although it is often used as a single user's identity, this concept can also extend to other people who may be of interest to the user's situation, such as friends in a social network, people located nearby or other users of the same application.

- *What does the user do?* The "what" aspect refers to the activity, another essential category of the Dey and Abowd approach and a crucial one too many Context models.

- *Where is the user located?* Where is the device or object? The location is by far the most popular aspect of Context.

- *When does the user perform a specific action? For how long?* Temporal Context has recently been explored to search for models and routines in people's daily lives. The temporal aspect is often linked to changes in the other categories that characterize the Context.

- *Why does the user take a specific action? Why is the user here?* Like the "when" aspect, the "why" aspect is related to other categories. For example, one wonders: *Why this activity or position?* This information is perhaps the most complicated aspect of the Context to be analysed since one must consider the meaning of the action, intention or emotion. The emotional Context, in particular, is not easy to interpret.

A thorough understanding of Context is essential to choose or design a suitable model. In this regard, a uniform approach is required to represent information associated with the Context and Context itself.

1.1.3 Context Representation

Effectively representing an abstract concept such as Context can be almost as complicated as providing a precise definition. In the *context-aware projects* developed over the years, different models have been proposed and used for trying to quantitatively capture the relevant characteristics of the Context to make them available to the following processing stages. Although these solutions are usually designed for a particular scope and use very different approaches, it has been possible to identify some characteristics they share:

- The Context is decomposed into dimensions or attributes that identify the relevant system characteristics, such as time, position and temperature.
- The various dimensions are made measurable by defining quantities, units of measurement and ranges of permissible values.

A model called "Context Spaces" associates to each relevant feature of the Context - "context attribute" - a dimension, thus shaping a multi-dimensional space that represents all the possible situations in which the system can be found (Dominici *et al.*, 2012). The current Context, defined by the values assumed by the attributes, can be represented as a geometric point in this space (Figure 1.2): the axes identify the dimensions; the dotted line shows the evolution of the system from one Context to another; the subspaces highlight some relevant configurations among the possible ones. In addition to being very intuitive, this model allows the use of geometric instruments such as trajectories and distances to reasoning and operations on the system. One of the significant limitations of this approach, which affects its expressive capacity, consists in the impossibility of ordering the dimensions in a hierarchical way or with a topology, in stark contrast with the ontological models and, in general, the Knowledge Organization Systems (KOS).

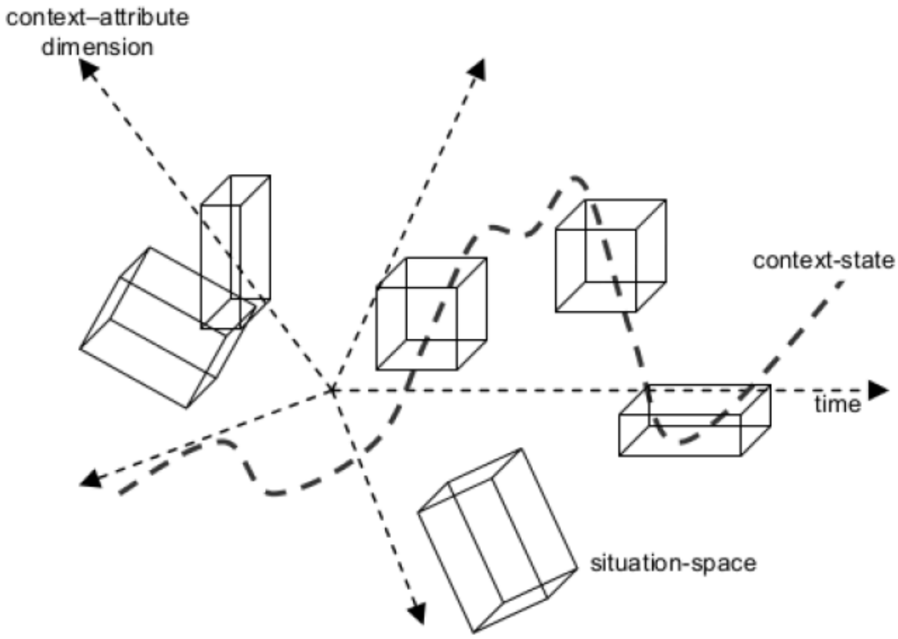

Figure 1.2 *Representation of the "Context Spaces" Model*

Another model, widely used and of great interest, is the Context Dimension Tree (Rauseo *et al.*, 2013); an example is shown in Figure 1.3. The elements that make up a CDT are:

- The **root node** represents the set of possible Contexts described by the CDT.

- **Dimension nodes** represent the relevant characteristics of the Context, such as time, position and topic of interest (represented by black circles).

- **Value nodes** (or **concept nodes**) identify the possible values of the dimension to which they are connected, in discrete and not too numerous domains (represented by white circles).

- **Parametric nodes** identify the value of the dimension to which they are connected when there are continuous or too numerous domains (represented by triangles or squares, respectively, connected to dimension nodes or value nodes).

- **Constraints**: highlight the prohibited combinations of nodes in the tree because they are combined with Contexts without real meaning.

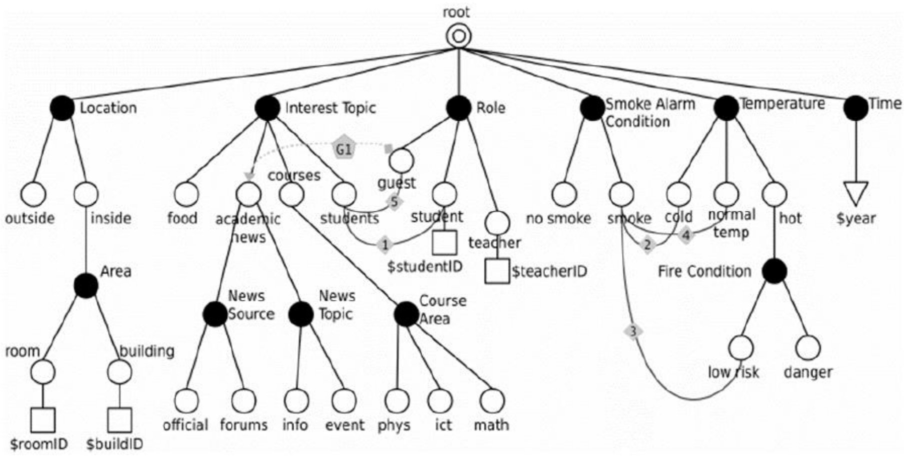

Figure 1.3 *Example of Context Dimension Tree applied to a simple scholastic domain*

The current Context, defined by the various dimensions' values, can be represented as a subgraph. Each dimension node is combined with at most a value or parametric node, and the Context is defined as an AND between different Context elements.

Therefore, adopting a hierarchical structure allows, besides orthogonally separating the various dimensions of the Context, to use different abstraction levels to specify and represent all the possible and eligible Contexts in a given application domain.

1.1.3.1 Context Dimension Tree

The Context Dimension Tree is a tree consisting of a triad $< \mathrm{r};\ \mathrm{N};\ \mathrm{A} >$ where r indicates its root, N indicates the set of nodes of which it is composed and A indicates the set of bows that connect these nodes (Orsi and Tanca, 2011; Rauseo *et al.*, 2013). The CDT is used to represent graphically every possible Context within an application.

The nodes are divided into two categories: dimension nodes and concept (value) nodes. A dimension node, which is graphically represented with black colour, describes a possible dimension of the application domain; a concept node is represented instead with white colour and represents one of the possible values that a dimension can assume. Every node is identified using its type and a label.

The child nodes of the root node r are all dimension nodes and are referred to as "top dimension". For every top dimension, there can be a subtree: a dimension node can only have concept nodes as child nodes; similarly, a concept node can only have dimension nodes as child nodes; the leaves nodes have to be instead concept nodes or parametric nodes.

The introduction of parameters is due to their utility in modelling some features which could have an infinite, or at least a high number, of attributes (Figure 1.4). For example, a node indicating the Cost dimension of a museum ticket would risk having an increased number of values which would need to be specified via relevant concept nodes children. Therefore, it is more comfortable, in such a case, to use just one parameter, whose value shall be defined case by case. The leaves nodes, other than concept nodes, can also be parameters.

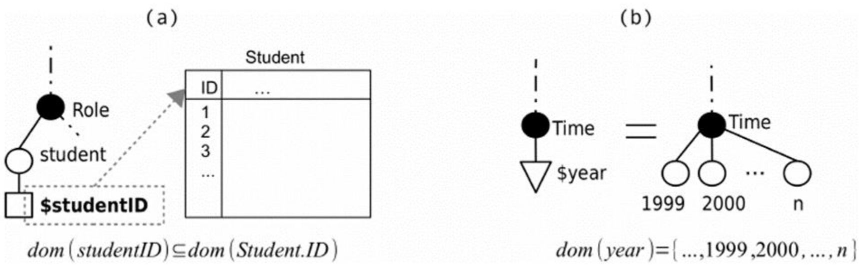

Figure 1.4 *Example of parameters linked to a concept node (a) and a dimension node (b)*

Generally speaking, every parameter node has a corresponding domain, *dom(nP)*. For the parameters nodes linked to concept nodes, the domain can be a set of key values from a relational database. In contrast, in the event of parameters nodes connected to dimension nodes, the domain is a set of possible concept nodes of the dimension.

Regarding the Context Dimension Tree, the concepts of "context element", "context", "constraint" and "partial and contextual views" can be defined, which would be particularly useful.

- A **Context Element** is defined as an assignment: d_{name_i} = value, where d_{name_i} indicates a possible dimension or under-dimension of the CDT (it is, therefore, a label of a dimension node). Simultaneously, the value could represent the label of one of the concept nodes under the dimension node considered, or the value of a parameter related to one of those concept nodes, or the value of a parameter referred to as the dimension node at stake. As an example, the following assignments are possible Context Elements: $Interest = Free\ classrooms$; $Locality = Area(Id = 3)$; $Interface = Smartphone$; $Role = Student(Id = 123456)$. Another example could be related to a tourism scenario: $Interest = Modern\ art$; $Locality = Museum(Id = 5)$; $Interface = Tablet$; $Role = Tourist(Id = 654321)$

- A **Context** is specified as: $\wedge\left(d_{name_i} = \text{ value}\right)$. Therefore, it is defined as an AND between different Context Elements: different Context Elements combined through an AND give life to a Context. For example, a possible Context that can be obtained from the listed Context Elements is: $\mathrm{C} \equiv (Interest \ = \ Free\ classrooms) \wedge$ $\left(\text{Locality} = \text{ Area(Id } = \text{ 3)}\right) \wedge (\text{Interface } = \text{ Smartphone }) \wedge$ $\left(\text{Role } = \text{ Student(Id } = \text{ 123456)}\right)$. Using the Context Dimension Tree, after having analysed the application domain, it is possible to graphically express its characteristics, dimensions and values that they can assume through dimension nodes and concept nodes or parameters. The assignment of one of the possible values to a dimension represents a Context Element; the Context Elements can be considered as the unique elements of the application through which a Context can be decomposed. When defining a Context, it is necessary to specify every Context Element that contributes to its creation. Every Context can be expressed in terms of the combination with AND of its particular Context Elements. From the given definition, it is possible to understand how views will be created on a database: they will be built up indeed starting from the portions of the database, hence from the partial views associated with the Context Elements taking part in Context formation.

- Some **constraints** would be appropriate not to violate when defining a Context through the CDT. In a Context C defined as one AND between different Context Elements, i.e. $\wedge\,(d_{name_i} = \text{value})$, it must be ensured that, between them, there are no Context Elements that are:
  - siblings nodes with the same dimension node as a father, hence are two Context Elements like $d_{name_a} = value_1 \ \ and \ \ d_{name_a} = value_2$ (*validity property 1*);
  - descended from the same ancestor, as long as the root node is not considered, in this case, as the ancestor node (*validity property 1*);
  - descended from each other (*validity property 2*).

Considering Figure 1.5, a possible couple of Context Elements that cannot appear in the same Context is shown. The couple is the one composed by $Dim_a \ = \ Conc_{a,1} \ and \ Dim_c \ = \ Conc_{c,1}$ since they both descend from the same ancestor dimension node Dim_a (*property 1*).

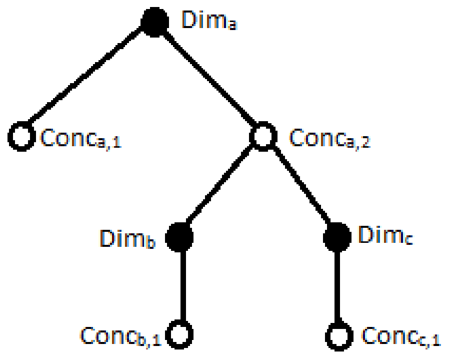

Figure 1.5 *Context Elements: descendants from the same ancestor*

Always considering Figure 1.5, an incompatible couple of Context Elements is the one composed by $Dim_a = Conc_{a,2}$ and $Dim_c = Conc_{c,1}$, since the last one stems from the previous Context Element.

Examples of the Contexts and the relevant prospects extracted from the Context Dimension Tree in Figure 1.3 are:

- "student < s100878 >, outside, food": the student, identified by studentID=s100878, is off-campus and looking for a place to eat;
- "teacher, room, official": a lecturer in a classroom is looking for official news.

Concerning a CDT for a university application scenario, the Context "student, lecturer" represents a situation without meaning, violating the *validity property 1*. The designer thought of students and lecturers as users with access to data with different roles; hence different Contexts must be defined. On the contrary, the two previous Contexts examples describe significant situations. Moreover, since, for example, the information about events properly contain all news, Contexts that include both academic news and events (redundancy) should also be discarded. This one falls under the *validity property 2*.

In Figure 1.6, some examples are provided of Contexts that breach the validity properties. The Context "inside, outside" infringes the *validity property 1* because they are thought to be a different description of the same perspective. Similarly, the Context "outside, room < anyID > " violates the *validity property 1*. Also, since the information concerning "inside" contains all the data concerning "building" and "room", Contexts containing both "room < anyID > " and "inside" should be excluded since it would be redundant and violates the *validity property 2*.

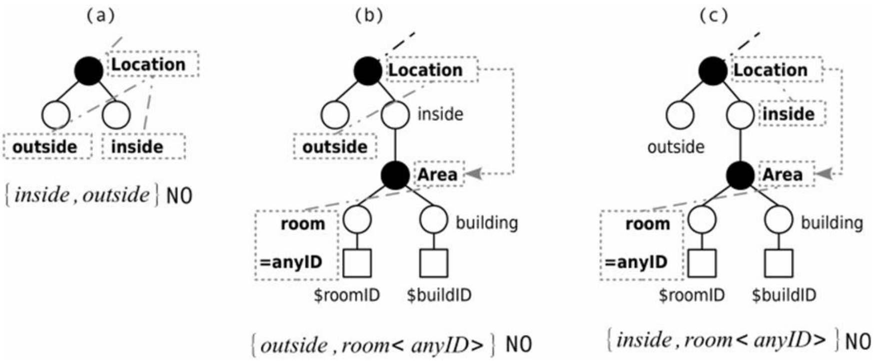

Figure 1.6 *Examples of violation of validity properties*

Once designed the Context Dimension Tree and defined all the Context Elements, one can notice that some of these, combined, lead to the creation of Contexts lacking relevance with a view to the application considered: a**pplication constraints**.

In order to forbid the combination of such Context Elements, the declaration and the specification of some constraints are required.

There are many types of restrictions; in particular, three of them are worth an analysis: ***useless-context constraints***, ***dimension-independent constraints,*** and ***preferred-detail constraints***.

In the CDT in Figure 1.3, an example of Useless-Context Constraint is the one between Smoke and Low risk that cannot coexist in the same Context (for example, the rules of procedure could define that the fire alarms should not be activated when the risk is low). The other example is between the values guest (Role) and student (Interest Topic) because the guest role should not have access to students' data (Figure 1.7). Graphically the Useless-Context Constraints are specified on the CDT with lines that link couples of white nodes but only for the values couples forbidden within the same Context. In contrast, more complex constraints are not shown in the figure.

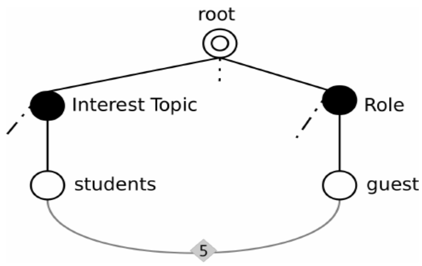

Figure 1.7 *Example of useless-context constraint*

Instead, Dimension-Independent Constraints are used to shape and prevent situations where, given a Context, adding the value of a dimension (i.e. the inclusion of a Context Element), in the beginning, was not considered, does not change the global view of the Context. Hence, the specification of such value is irrelevant to the final view. For example, it is possible to understand how the Context related to a lecturer's $Salary$ would not change if the Locality Context Element was added.

Finally, a Preferred-Detail Constraint is used to specify the preferred level of detail regarding the information. The constraints on the degree of granularity are defined in order to limit the level of description of a specific Context instance, referred to as a *master instance*. These are represented on the CDT through arches, marked by a pentagon, connecting nodes constituting Context Elements in the master instance and a node corresponding to the restricting Context (Figure 1.8).

A **Doable Context** is an instance of semantically valid Context, i.e. it does not break any constraints depending on the application in the considered application domain.

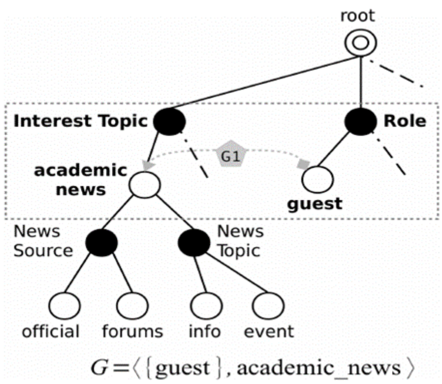

Figure 1.8 *Example of preferred-detail constraint*

After defining all the Contexts and all their Context Elements, a different portion of the database is associated with every Context Element, containing its relevant data.

The **partial views** (p_{view}) are defined in tight association with the development of the Context Dimension Tree to which they relate. Every partial view is linked to a particular concept node of the CDT, thus to the Context Elements built starting from these nodes. It returns a part (small if possible) of source data that have been detected as relevant for this perspective by the designer. Similarly, the dimensions with parameters have partial views linked to the parameters' values since the parameters linked to a dimension represent shortcuts to concept nodes.

By the Context Dimension Tree structure, the Context Elements built from nodes located at different levels provide different abstraction levels. The partial view associated with those Context Elements considers different levels of granularity in terms of data filtering. In general, the partial views concerning Context Elements established starting from top-level will include, often strictly, partial views built starting from lower levels. The concept nodes with parameters have partial parametric views, providing a data filtering system accurate in the associated relation.

After defining the associated partial views to every Context Element, it is necessary to obtain a final view of the Contexts.

A Context is generally composed of many Context Elements; that being said, one needs to specify a method to combine the different partial views corresponding to such Context Elements. The **contextual views** provide the combination mechanism (c_{view}). These global views are composed upon the partial views to work on a higher level of representation: the contextual views provide the real mapping of the source database to a contextual database that contains the data relative to the instance of Context.

1.1.4 Context-Aware Systems

As for the definition of Context, Context-Aware Computing was also treated and introduced by Schilit and Theimer in 1994 (Schilit *et al.*, 1995): «Such context-aware software adapts according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time».

There have been several attempts to define Context-Aware Computing from the previous definition, but most of them have been too specific and too challenging to be put into practice.

In this regard, Dey provides a more general and practical definition (Dey, 2001): «A System is context-aware if it uses context to provide relevant information or services to the user, where relevancy depends on the user's task».

Compared to the previous definition, the difference lies in that an application to be considered *context-aware* does not necessarily need to perceive the whole Context but must react to it, giving appropriate answers to the user regarding information and services.

1.2 Related Works

The Internet has allowed us to create a robust information network through which more and more services are spread: from information to communication, from banking services to purchasing. It has also allowed us to connect human beings to communicate and instantly share anything anywhere. In this sense, in 1999, Kevin Ashton defined the term Internet of Things (IoT) (Ashton, 2009) which refers to the concept of a network in which human beings and machines are connected using standard public services.

The Internet of Things is a possible evolution of the network's use: the objects ("things") will make recognizable and acquire intelligence thanks to communicating information about themselves and access aggregated data from others: alarms ring before in case of traffic, shoes transmit time, speed and distance to compete in real-time with people on the other side of the globe, medicine boxes alert family members if you forget to take the right dose of medicine. All objects can acquire an active role through the connection to the Network.

One of the Internet of things' goal is to ensure that the electronic world draws a map of the real one, giving electronic identity to the physical environment's things and places. Objects and places equipped with Label Radio Frequency Identification (RFID) or QR Codes communicate network information. The fields of application are manifold: from industrial applications (production processes), logistics and mobile information, up to efficiency energy, remote service and environmental protection. In the vision of the Internet of things, objects create a pervasive and interconnected system using multiple communication technologies (typically short range).

According to Atzori (Atzori *et al.*, 2010), the IoT idea's main strength is the high impact on several aspects of potential users' everyday life and behaviour. The spread of low-cost sensors makes a significant contribution to creating an impressive amount of data. Moreover, thanks to their pervasiveness, they appear able to influence peoples' daily actions increasingly.

In the last decade, the Internet of Things (IoT) paradigm embraces Context Awareness as a core feature of smart and pervasive systems (Piccialli and Chianese, 2017). One of the fields that can take significant advantages of such paradigms is Cultural Heritage (Colace, De Santo, Greco, Lemma, *et al.*, 2014). Collection, modelling, reasoning, and distribution of Context related to sensor data play a critical role in this challenge. In fact, Context-Aware Computing has proven to be successful in understanding sensor data (Perera *et al.*, 2014).

For example, e-SENSE (Gluhak and Schott, 2007) enables ambient intelligence using wireless multi-sensor networks to make context-rich information available to applications and services. e-SENSE combines Body Sensor Networks (BSN), Object Sensor Networks (OSN), and Environmental Sensor Networks (ESN) to capture the Context in the IoT paradigm. The features required by Context-Aware IoT middleware solutions are identified as *sensor data capturing*, *data pre-filtering*, *context abstraction*, *data source integration*, *context extraction*, *rule engine*, and *adaptation*.

A further example is Hydra[3] (Badii *et al.*, 2010). It is an IoT middleware that aims to integrate wireless devices and sensors into ambient intelligence systems. Hydra comprises a Context-Aware Framework (CAF) that provides high-level, powerful reasoning capabilities based on ontologies and lower-level semantic processing based on an object-oriented/key-value approach. The CAF contains two core components: the Data Acquisition Component

(DAqC) and the Context Manager (CM). DAqC is responsible for connecting and retrieving data from sensors. CM is liable for Context Management, Context Awareness and Context Interpretation. CAF models three distinct types of Context: Device Contexts (e.g., data source), Semantic Contexts (e.g. location, environment and entity) and Application Contexts (e.g., specific domain). Hydra identifies *context reasoning rule engine*, *context storage*, *context querying* and e*vent/action management* as the Context-Aware Framework's key components.

As shown below, the tourism domain and, in particular, Cultural Heritage can benefit from applying Context-Aware Intelligent Systems within spaces such as museums, temporary art exhibition, monuments, and buildings to enhance the people enjoyment and the promotion of such areas. An example is SmartApp Salerno (Colace, Lemma, *et al.*, 2017): a Context-Aware App for e-tourism. The App collects information from social environments adapting the proposed itinerary taking into account the communities and the user's interests. The entire approach has been tested inside the town of Salerno with exciting results.

In this respect, during this research work, an adaptive Context-Aware System will be implemented. It will collect not-structured data belonging to heterogeneous sources and develop tailored recommendations for the user to support tourists. The solution found will have to take advantage of information technologies, like Recommendation Systems, and the objective will be achieved using a Context-Aware Architecture.

Chapter 2
Recommender Systems

Due to modern information and communication technologies (ICT), it is always easier to exchange data and have new services available through the Internet. However, the amount of data and services available increases the difficulty of finding what one needs.

In this scenario, a Recommender System (RS) represents one of the most promising solutions to overcome the so-called information overload problem, analysing users' needs and preferences.

RS are applied in different sectors but have one goal: to help people make choices based on an analysis of their behaviour or of that of users similar in terms of characteristics or interests. In other words, these systems are widely used to analyse and filter information in support of an individual's choice of a service or a specific object.

The Recommender Systems developed considerably in the nineties thanks to the beginning of the first e-commerce sites, solving the problem of managing a considerable amount of data. They then adapted to the technologies developed over the years to improve their performance. Just think, for example, of the birth of social networks. In this context, RS can take advantage of user groups' opinions belonging to the same community (Eirinaki *et al.*, 2018). In particular, these systems' importance significantly increased in the last decade, representing a massive opportunity in the modern context characterized by Big Data (Castiglione *et al.*, 2018). By processing large quantities of data and the use of increasingly sophisticated algorithms, the ultimate goal is to provide precise suggestions to users (Portugal *et al.*, 2018).

The field of use of Recommender Systems is very varied. Its use is known above all in e-commerce, in streaming services and dissemination sites, or in cases where many services or objects are made available (regardless of whether they are books, clothes or films). It is because only a part of them is interesting or relevant to the user. Among the main RS types, there are content-based systems, collaborative systems and hybrid systems (Bobadilla *et al.*, 2013). Added to these are also advanced recommendation services that manage and use the entire Context in which a user is located: Recommender Systems based on Context-Aware Technologies (Casillo, Clarizia, *et al.*, 2019).

As previously mentioned, one of the main characteristics of Recommender Systems is to predict the consideration that an individual may have about an item that has not yet been evaluated. How this forecast is made is one of the distinguishing criteria for the RS. It should also be added that the ability to predict ratings correctly distinguishes a reliable Recommender System from an ineffective one.

2.1 Background

The development of the Internet and the growth of interest in the e-commerce field and the new digital platforms have increased interest in Recommender Systems. RS is an information analysis and filtering tool that support a user in choosing an item (Colace, De Santo, *et al.*, 2017). The main elements on which a Recommender System operates are *user, item,* and *transaction* (Ricci *et al.*, 2015).

- The *user* represents the recommender phase's target, which can be identified through its needs and characteristics.

- The *item* represents an element that Recommender System suggests to the user. It can be classified according to its features. In Recommender Systems, it is fundamental to understand how an item feature influences its usefulness for a given user.

- The *transaction* represents the interaction between the system and the user. In particular, during the system's use, helpful information is stored to generate recommendations. This information can be obtained explicitly, for example, through questionnaires or registrations, or implicitly by examining its dynamic behaviour over time.

2.1.1 Definition of Rating

The most common information exchanged during a user/system interaction is the rating: evaluating the user's consideration of an item. The rating can take various forms: the main ones being *Numerical Rating*, in which the user expresses his preference by choosing a range varying (for example, from one to five stars), *Ordinary Rating* in which the user selects a term following his opinion (Strongly agree, Agree, Neutral, Disagree, Strongly disagree), *Binary*

Rating (Like, Dislike), in which the user chooses whether he likes or not an item and, at last, *Unary Rating* that indicates whether the user has seen or selected an item.

The lack of evaluation indicates the lack of information about the relationship between user and item. A quantitative numerical representation is attributed to the user's usefulness or could assign to an item through the rating (Chang *et al.*, 2015). The concept of rating is formalized below.

Definition 2.1: *Let U be the set of users and I the set of items. The rating function or utility function is defined as the function r which to each pair of the domain $(u,i) \in U \times I$ associates the evaluation $r_{ui} \in R$.*

$$\mathrm{r}: (\mathrm{u}, \mathrm{i}) \in \mathrm{U} \times \mathrm{I} \mapsto \mathrm{r_{ui}} \in \mathrm{R} \tag{2.1}$$

Definition 2.2: *Let Z be a not empty set and let $f: Z \mapsto R$ be a function from Z to R, the argument of the maximum associated to f is defined as the function $arg\max f$ which returns the set of domain values that maximize the image of f.*

$$\arg\max_{\mathrm{x} \in \mathrm{Z}} \mathrm{f}(\mathrm{x}) = \{\mathrm{x} |\ \mathrm{f}(\mathrm{y}) \leq \mathrm{f}(\mathrm{x})\ \forall \mathrm{y} \in \mathrm{Z}\} \tag{2.2}$$

One of the main aims of a Recommender System is to determine the item $i' \in I$ which maximises the rating function for the user $u \in U$:

$$i'_u = arg \max_{i \in I} r(u, i) \tag{2.3}$$

A problem, which is common to all Recommender Systems, is the low number of known assessments. Therefore, the need to provide an estimate to evaluations $\widehat{r_{ui}}$, for each element of the domain $U \times I$ on which the rating function is not defined, arises. As mentioned above, the ability to provide good forecasts distinguishes a right Recommender System from an ineffective one.

2.1.2 Recommendation Techniques

The main scope of Recommender Systems is to make adequate rating predictions on the pairs $(u,i) \in U \times I$ where the rating function is not defined. Based on how these forecasts are made and how information relating to users and items is stored and processed, it is possible to classify Recommender Systems. *Content-Based RS* (Ricci *et al.*, 2015) recommend items similar to what the user has preferred in the past. These techniques require a preliminary phase for constructing and representing a user profile and the objects' features. *Collaborative Filtering RS* (Koren *et al.*, 2009) suggest to the user items that are well valued by users with similar characteristics (Aberger *et al.*, 2009). In this way, Recommender Systems are entirely based on evaluating the interests expressed by users of a community. *Knowledge-Base RS* suggest an item to the user by considering how much the proposed item features meet the user's specific needs. It is essential to build an appropriate user profile (Ricci *et al.*, 2015) to determine these needs. *Demographic Filtering* recommends an item to the user by using the information regarding the region to which he belongs, the language, the gender and the age (Ricci *et al.*, 2015). *Community-Based RS* recommend an item to the user by using the information acquired from social networks. The means of acquisition are the user tags or their friends' preferences (Ricci *et al.*, 2015). *Hybrid RS* consists of combining two or more techniques from those listed so far (Figure 2.1). The commonly used techniques are Content-Based and Collaborative Filtering (Ricci *et al.*, 2015).

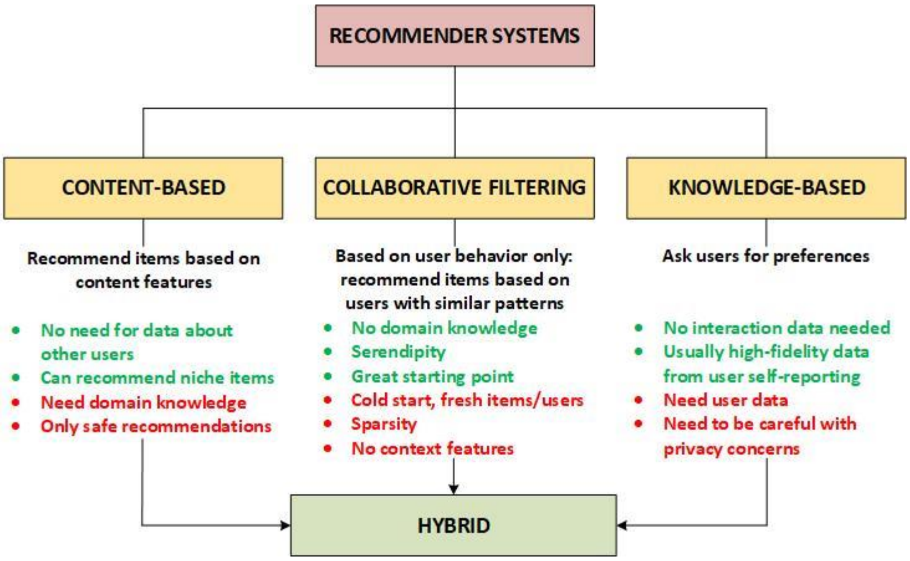

Figure 2.1 *Main recommendation techniques and some limits*

2.1.2.1 Content-Based Systems

Content-Based Recommender Systems generate rating forecasts through the vectors of items' features and the user's profile (user's preferences).

The structure of a Recommender System based on high-level content is composed of three modules.

In the *Content Analyzer*, the characteristics of the items are analysed to build a vector that describes them (Ricci *et al.*, 2015) and, afterwards, the *Profile Learner* involves the study of user preferences summarized through a numerical vector which constitutes the relative profile (Ricci *et al.*, 2015). At last, in the *Filtering Component*, the rating prediction $\widehat{r_{uj}}$ is generated through the cosine of similarity between the feature vector of the item $y_j \in R^n$ and the user profile vector $x_u \in R^n$:

$$\widehat{r_{uj}} = cos(x_u, y_j) = \frac{< x_u, y_j >}{\|x_{u_2}\| y_{j_2}} = \frac{\sum_{t=1}^{n} x_{u_t} y_{j_t}}{\sqrt{\sum_{t=1}^{n} x_{u_t}^2} \sqrt{\sum_{t=1}^{n} y_{j_t}^2}} \quad (2.4)$$

The user profile construction originates from the Information Retrieval to acquire information about the user's preferences (Portugal *et al.*, 2018). The main strength of the described recommendation technique consists of the system's ability to correctly learn user preferences through actions on different contents concerning the items for which the forecast is generated.

2.1.2.2 Collaborative Filtering Systems

Collaborative Filtering Systems are based on an ancient concept for humans: sharing opinions. In fact, predictions are made through a community's opinions (Geuens *et al.*, 2018).

Through the Internet, the trivial "word of mouth" is replaced, passing from the analysis of hundreds of opinions to thousands or more. The speed of the calculation systems allows to process the vast amount of information obtained in real-time and to determine both the preferences of a large community of people and the preferences of the individual through the most reasonable opinions for the specific user or a group of users (Francesco Colace, De Santo, Greco, *et al.*, 2015).

Collaborative Filtering (CF) generates rating forecasts based on ratings known to the system.

Collaborative Filtering Systems are subdivided into two groups:

- *Memory-Based CF* is divided into three main strategies:
    - *User-Based CF*: users are divided into groups, called *neighbourhood* (F. Colace *et al.*, 2015; Zhang *et al.*, 2016). The similarity can be calculated through the Cosine Vector or the Person Correlation (Desrosiers and Karypis, 2011).
    - *Item-Based CF* creates groups of items through the Person Correlation for the items or the Adjusted Cosine formula (Desrosiers and Karypis, 2011).
    - *User-Item Based CF* combines User-Based CF and Item-Based techniques to overcome the limits of individual techniques.

- *Model-Based CF:* the aim is to create a *model* by factorising the matrix of available ratings. The most common methods for this are Principal Component Analysis (PCA) (Bokde *et al.*, 2015), Probabilistic Matrix Factorization (PMF) (Mnih and Salakhutdinov, 2007), Non-Negative Matrix Factorization (NNMF) (Bokde *et al.*, 2015) and Singular Value Decomposition (SVD) (Symeonidis and Zioupos, 2016).

2.1.2.3 Hybrid System

The recommendation process's effectiveness can be improved by combining, for example, Content-Based strategies and Collaborative Filtering (Ranjbar Kermany and Alizadeh, 2017). In this way, the limits of the individual techniques can be overcome.

A hybrid system can be achieved by using collaborative and content-based methods separately. Then it is possible to combine forecasts, incorporating some features of a content-based system in a collaborative approach or vice versa, through the construction of a single model which includes both techniques.

Hybrid approaches combine the two recommendation techniques to exploit one's advantages and to correct the other's disadvantages (Paradarami *et al.*, 2017).

Table 2.1 shows the main advantages and disadvantages of the different recommendation techniques.

Table 2.1 *Summary of the advantages and disadvantages of the main recommendation techniques*

METHOD	TECHNIQUE	ADVANTAGES	DISADVANTAGES
Content-Based Filtering		✓It does not require many users for forecasting ✓It requires knowledge of the features about items only	- Limited content analysis (no serendipity) - Over-specialization: the system tends to suggest the same type of item - Cold start (new user)
Collaborative Filtering	*Memory-Based*	✓Ease of implementation and data addition ✓It adapts well to related items ✓It offers serendipity of results	- It depends on the present ratings - Cold start (new user / new item / new community) - Limited scalability for large datasets - It works poorly with sparse matrices
	Model-Based	✓It works well with sparse matrices ✓It adequately addresses the scalability problem	- High cost for the construction of the model - Loss of information
Hybrid Recommender System		✓It takes advantage of the advantages of different techniques ✓It interacts well with the issues of the techniques that compose it	- High cost for the construction of the model - Increase of complexity

2.1.3 Some limits of Recommender Systems

Recommender Systems are faced with several practical issues. Among these, the main ones are *scalability*, which indicates the Recommender Systems' ability to adapt to increases in the number of data to be managed, and sparse rating matrix, which indicates a small number of available ratings. The latter implies the need to have a Recommender System, which can provide suggestions based on a limited number of available evaluations. Finally, the Cold Start problem indicates the difficulty of a Recommender System to provide suggestions to a new user or a new item (Karabadji *et al.*, 2018).

2.2 Related Works

Recommendation Systems must return a rating prediction, i.e. a numerical value corresponding to the user's consideration for a given item. For this purpose, the information available has to be analysed.

For example, in (Wang *et al.*, 2018), information about the abstract of a scientific article is acquired in order to suggest the most appropriate journals or conferences submit it. After choosing the feature acquisition mode, the Content-Based approach generates the rating predictions through the Softmax Regression, which generalises the Logistic Regression. Instead, in (Son and Kim, 2018), a Recommender System is developed to suggest exciting articles. The recommendation is made through a multilevel simultaneous citation network that allows considering the information of interest related to the article itself.

In (Albanese *et al.*, 2011), Recommendation Systems are applied in the field of Cultural Heritage. Specifically, an application for browsing the Uffizi Gallery digital picture collection is designed.

If some ratings are known, it is possible to proceed with a Collaborative Filtering approach. An example of Memory-Based CF can be found in (He *et al.*, 2018), where an Item-Based CF is built through a Neural Network. In particular, importance is given to the user's history.

In (Chen *et al.*, 2021), a Neighbourhood-Based Recommendation method is developed by integrating the kernel function to calculate the similarity between the entities considered. A further example is given by (Wang *et al.*, 2021), where the similarity between the entities is differently calculated since it is obtained through alpha-divergence.

In (Yassine *et al.*, 2021), the recommendation process involves grouping items according to genre attributes selections. A user profile is then generated to determine the user's preferred movie class. At this point, rating predictions are generated only for movies belonging to the user's preferred movie class.

A case of applying probabilistic matrix factorization and a Model-Based Recommender System is provided by (Mnih and Salakhutdinov, 2007). The PMF application to the Gradient Descent provides a stable Recommender System in the presence of a few users' ratings.

Finally, among the possible Model-Based approaches that exploit Singular Value Decomposition (SVD), the Average Rating Filling (ARF), Stochastic Gradient Descent (SGD) and Biased Stochastic Gradient Descent (BSGD) methods are widely used (R. Wang *et al.*, 2019; Zhao *et al.*, 2019).

Chapter 3
Context-Aware Recommender Systems

In the Big Data era, every sector has adapted to technological development to service the vast amount of information available.

In this scenario, systems able to recommend the right services play a crucial role and, in particular, Context Awareness helps improve the recommendations provided. Over the years, the concept of Context-Aware Recommender Systems (CARS) has been introduced to obtain increasingly reliable rating forecasts.

3.1 Background

The purpose of Recommender Systems (RS), as seen above, is to generate forecasts through the chosen strategy and information about users and items. In this regard, new approaches to developing a Recommender System have also considered *contextual information* that the system can use to predict rating values.

Contextual information can therefore be considered information that modifies the *transaction* mode of a Recommender System through the data obtained from the situation involving the system (user and item). Consequently, the rating value changes as contextual situations change. In this way, the Context can be defined as the specific situation defined by the Recommender Systems' contextual parameters.

The introduction of the Context within CARS changes the definition of the utility function or rating (Figure 3.1).

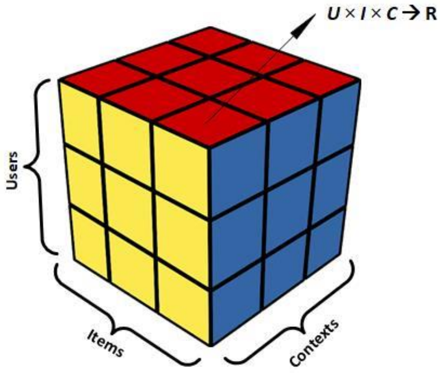

Figure 3.1 *Multi-dimensional space: Users, Items and Contexts*

Definition 2.1: *Let U be the set of users, I the set of items and C the set of contextual variables. The rating function or utility function for CARS is defined as the function f which to each (u, i, c) of the domain $U \times I \times C$ associates the evaluation $f_{uic} \in R$.*

$$f: (u, i, c) \in U \times I \times C \longmapsto f(u, i, c) = f_{uic} \in R \qquad (3.1)$$

The introduction of CARS shows the importance of the acquisition of contextual information. These can be obtained in various ways, such as:

- *explicitly*, by questions to the user or by other means (Shani and Gunawardana, 2011);
- *implicitly*, via mobile devices (location, temporal data, climatic data) or by changing the user's environmental conditions (Adomavicius and Tuzhilin, 2015);
- through statistical or data mining techniques.

3.1.1 Classification of Context-Aware Recommender Systems

Once the methods of acquiring the contextual information and their classification are defined, it is necessary to use the same ones inside the recommendation process.

The possibilities of insertion of contextual information within the Recommender System are divided into three approaches (Figure 3.2):

- **Contextual Pre-Filtering.** Information is applied to data before the development of recommendations. In this case, it can be assumed that the context is a function:

$$c: (u, i) \in U \times I \longmapsto c(u, i) = (\bar{u}, \bar{\imath}) \in U \times I \qquad (3.2)$$

such that the utility function r of the Recommender Systems is calculated on the image obtained by c:

$$r \colon (\bar{u}, \bar{\imath}) \in U \times I \longmapsto r_{ui} \in R \tag{3.3}$$

- **Contextual Post-Filtering.** Contextual information is initially ignored by proceeding with a classic recommendation approach: it will be used for the final filtering of the recommendations obtained. The context can be imagined as a function that modifies the value of the utility function r:

$$c \colon r_{ui} \in R \longmapsto \overline{r_{ui}} \tag{3.4}$$

  where the rating function is filtered through the information acquired.

- **Contextual Modelling.** Contextual information is integrated into the recommendation process by creating an integrated model in the RS. In particular, contextual information is integrated into the utility function for the calculation of recommendations.

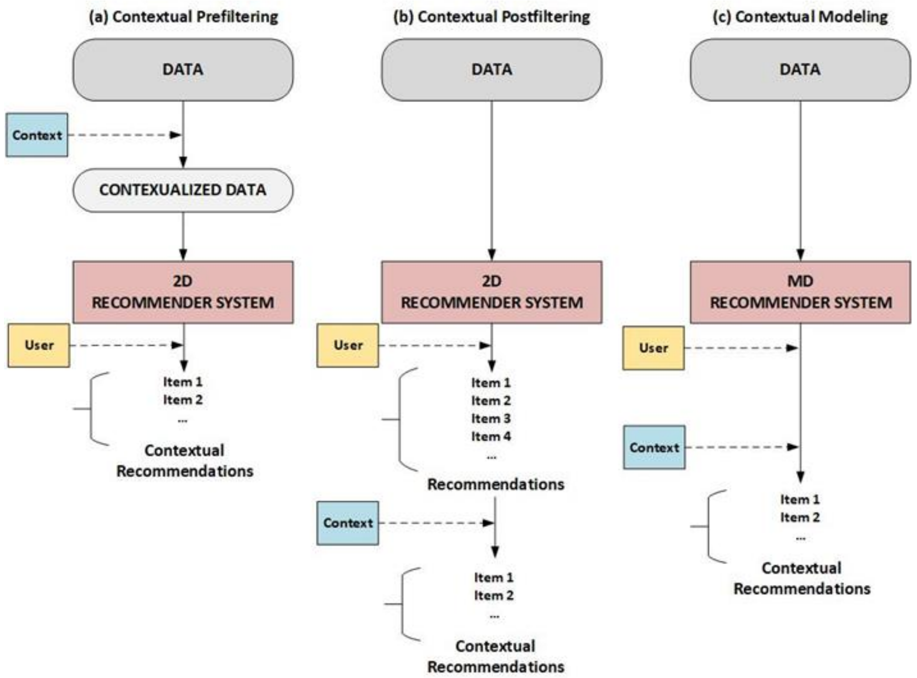

Figure 3.2 *CARS approaches*

The structure for the integration of contextual information about the two most common recommendation strategies is presented below:

- **Content-Based CARS**
  - *Pre-Filtering:* the context is integrated into the generation of user and item profiles (features' vectors). Contextual data is usually focused on user profiles. Such an approach previews the generation of many profiles according to the combinations of the concept nodes associated with the specific context's dimension.
  - *Post-Filtering:* the context data are used to filter the recommendations generated by the Recommender System. This approach provides for an additional number of calculations proportional to the dimensions of the elements of the sets U and I.
  - *Modelling:* contextual information is integrated into the computation of the *cosine similarity* between a specific user and item. This process generally involves heuristic formulas.
- **Collaborative Filtering CARS**
  - *Pre-Filtering:* the rating matrix may have additional columns or rows to the classical one. Other columns or rows are associated with different preferences in different contexts. This approach reduces computational complexity but requires the system to expend additional effort to acquire data.
  - *Post-Filtering:* as in the case of Content-Based CARS, contextual information is used to filter the generated rating predictions. This approach intuitively involves several additional calculations.
  - *Modelling*: this approach involves the integration of contextual data at the heart of the recommendation process. It is divided into:
    - *Heuristic-Based*: contextual information is integrated for determining Neighborhoods (analogous to the *cosine similarity* in Content-Based CARS).
    - *Model-Based*: tensor factorization is exploited by integrating contextual data into additional dimensions. This procedure aims

to find a numerical structure of the problem, which includes the context.

In addition to these classification groups, there are techniques, such as Context-Aware Matrix Factorization (CAMF), classified as Modelling Techniques, that integrate contextual information within the calculation of forecast obtained through the matrix factorization.

3.2 Related Works

Over the years, the study of Recommender System and Context-Aware Recommender Systems gave rise to numerous operational strategies, which eventually led to classifications seen before. Some challenging strategies will be shown below.

3.2.1 Content-Based

An interesting proposal of a Content-Based Technique is proposed by LOOKER (Missaoui *et al.*, 2019). This work proposes a pre-filtering strategy for introducing contextual information: a mobile Recommender System aimed to provide tourism and travel-related services.

This technique consists of two modules:

- *Spatio-Temporal Filtering Module* aims to select only the relevant items based on contextual information related to time and physical location.
- *Content-Based Filtering Module*, which provides rating forecasts. In turn, it is divided into three main components:
  - Multi-layer User Profile: through Statistical language modelling (Ponte and Croft, 1998), the comments of each user, within the categories "tourism services", are analysed. In particular, for each category c, $R_c = \{r_i, i = 1, \dots, \mathrm{tot}_c\}$ is the set of positive user reviews about the category under consideration and the component θ_c of the user profile θ is calculated as follows:

$$\theta_c = \frac{1}{|R_c|} \sum_{r_i \in R_c} P\,(w|r_i) \qquad (3.5)$$

where w is a word present in a subset of R_c and $P(w|r_i)$ is estimated by the Dirichlet Prior Smoothing (Zhai and Lafferty, 2004).

- TR-Services Profile: the service profiles are generated in a similar way as for users. Defined the set R_s of positive reviews of users about the item s, the component ω_s, associated with the corresponding category of the ω profile, is obtained as follows:

$$\omega_s = \frac{1}{|R_s|} \sum_{r_j \in R_s} P\left(w \middle| r_j\right) \tag{3.6}$$

- Content-Based Filtering Algorithm: rating forecasts for each user u, service s and category c are calculated using the following formula:

$$\overline{r_{u,s,c}} = \frac{1}{D_{KL}(\theta_c \| \omega_s)} \tag{3.7}$$

where $D_{KL}(\theta_c \| \omega_s)$ is the Kullback-Leibler divergence.

As part of the Content-Based Recommender Techniques, Hong et al. (Hong *et al.*, 2009) provide a different integration of contextual information proposing a framework based on a decision tree algorithm to insert the context through modelling strategy. The method is developed in four stages using a multi-agent system:

1. Data Gathering Layer: the user profile is generated and the services are selected. The context row is generated through wrappers.

2. Context Management Layer: context aggregator collects contextual information in a vector then transformed into a high-level context by context inference agent. The user profile and the selected services from the first phase united to the high-level context form the context history further elaborated through filtering agent.

3. Preference Management Layer: the decision tree algorithm is used to extract user preferences for each service. The association agent deduces the association rules for the selected services.

4. Application Layer: personalised services are provided.

Another example of a Content-Based Recommender System is provided by Shin et al. (Shin *et al.*, 2009), which presents a Context-Modelling integration. This approach develops as follows:

1. Calculation of vectors integrated with the context:
   - $u_i = < uc_{i1}, \dots, uc_{ij}, \dots, uc_{iq} > \in R^q$, correlation vector between the user i and q contextual information considered.
   - $s_k = < sc_{k1}, \dots, sc_{kj}, \dots, sc_{kq} > \in R^q$, correlation vector between the item k and q contextual information considered.
   - $h_i = \left(< hcc_{i1}, \dots, hcc_{iq} >, < hcd_{i1}, \dots, hcd_{iq} >\right) \in R^{2q}$, the vector that stores the contextual information.
2. Are computed:
   - $f_{UH}(i,j) = \frac{\sum_{x=1}^{q} |uc_{ix}| \times |hcc_{jx}|}{\sqrt{\sum_{x=1}^{q} |uc_{ix}|^2} \sqrt{\sum_{x=1}^{q} |hcc_{jx}|^2}}$, matching between user i and context history j.
   - $f_{HI}(j,k) = \frac{\sum_{x=1}^{q} |hcd_{jx}| \times |sc_{kx}|}{\sqrt{\sum_{x=1}^{q} |hcd_{jx}|^2} \sqrt{\sum_{x=1}^{q} |sc_{kx}|^2}}$, matching between context history j and item k.
3. Calculation of the rating forecast with the integrated time context: $f_{UI}(i,k) = \sum_{j=1}^{l} f_{UH}(i,j) f_{HI}(j,k)$

Colombo-M. et al. (Colombo-Mendoza *et al.*, 2015) proposes a Content-Based Recommendation Model with hybrid integration of Pre-filtering and Post-filtering of contextual elements (*location*, *crowd*, *time*). "Location" represents the distance between the user and the considered target. "Crowd" considers the preferences of the active user and rating forecasts made on other users. "Time" intended as time available to the user and associated with the user profile is the time needed to access the service (evaluated on three modes of transport). The framework of Colombo-M. is as follows:

1) Update of the user profile
2) Pre-filtering associated with Time (step 1): the items that are not accessible are discarded.
3) Pre-filtering associated with the Location: items too far away are discarded through a specific Distance-Decay function.
4) Pre-filtering associated with Time (step 2): items are discarded according to information obtained from the previous point.
5) Calculation of Similarity
6) Collection of the information associated with the considered user

7) Calculation of the rating forecast
8) Post-filtering associated with Location and Crowd: calculated ratings are changed based on contextual information related to the context obtained by considering crowd and location.

Table 3.1 shows a summary of Content-Based Approaches presented above.

Table 3.1 *CARS based on Content-Based Approaches: summary table of some related works*

	Content-Based CARS			
	Pre-filtering	**Post-filtering**	**Context modelling**	***Contextual information***
Missaoui *et al.*, 2019	X			*Location; Time*
(Hong *et al.*, 2009)			X	*Activity; Human; Time*
Shin *et al.*, 2009			X	*Time*
Colombo-Mendoza *et al.*, 2015	X	X		*Location; Time*

3.2.2 Collaborative Filtering

Baltrunas and Ricci (Baltrunas and Ricci, 2009, 2014) fit into the Pre-filtering integration approaches. Their *Item Splitting* technique, in a system of m users and n items, applies to the columns of the $R \in R^{m \times n}$ rating matrix and changes it according to the different contexts. In particular, the rating column is split into two columns: the set of values assumed by the context C is given, the column of the rating matrix is split over the value $c_j \in C$, which mostly alters the known ratings. This approach results in a matrix of contextualised ratings $\hat{R} \in R^{m \times (n+l)}$ with l number of columns that have suffered the split. At this point, it is possible to proceed by using a classic Recommender System. It is stressed that R matrix is broken down based on a single contextual dimension and merging the remaining to preserve a linear time complexity (Figure 3.3). Therefore, a disadvantage of this approach is determining the contextual dimension based on which to split.

$$\begin{pmatrix} r_{1i} \\ \vdots \\ r_{mi} \end{pmatrix}$$

$$\swarrow \qquad\qquad \searrow$$

$$\begin{array}{ccc} \begin{pmatrix} r_{1,i,c_j} \\ \vdots \\ r_{m,i,c_j} \end{pmatrix} & \qquad\qquad & \begin{pmatrix} r_{1,i,c} \\ \vdots \\ r_{m,i,c} \end{pmatrix} \\ \textbf{if } c = c_j & & \textbf{if } c \in C - \{c_j\} \end{array}$$

Figure 3.3 *Item Splitting approach*

Figure 3.4 shows an example of the Item Splitting approach in which the nature of an item, from the user's point of view, may change in different contextual conditions (values of contextual dimensions). Hence, the item may consider as multiple items: one for each contextual condition.

User	Movie	Rating	Time	Location	Companion
U1	M1	3	Weekend	Home	Friend
U1	M1	5	Weekend	Theater	Spouse
U1	M1	?	Weekday	Home	Family

Assume Location (Home vs. Theater) is the best split condition

User	Item	Rating
U1	M11	3
U1	M12	5
U1	M11	?

M11: M1 seen at home; M12 = M1 seen not at home

Figure 3.4 *Example of Item Splitting approach*

Similarly to the procedure described above, as shown in Figure 3.5, it is possible to split the rating matrix's rows using the *User Splitting* technique (Said *et al.*, 2011). It may be helpful to consider one user as multiple users if they demonstrate significantly different preferences in different contexts.

User	Movie	Rating	Time	Location	Companion
U1	M1	3	Weekend	Home	Friend
U1	M1	5	Weekend	Theater	Alone
U1	M1	?	Weekday	Home	Family

Assume Companion (Family vs. Non-Family) is the best split condition

User	Item	Rating
U12	M1	3
U12	M1	5
U11	M1	?

U11: U1 saw the movie with family; U12 = U1 saw the movie alone or with a friend

Figure 3.5 *Example of User Splitting approach*

It is also possible to operate on both the columns and rows of the matrix according to the *User Item Splitting* technique (Zheng *et al.*, 2013): the process is simply an application of item splitting followed by user splitting on the resulting output. As *Item Splitting*, *User Splitting* and *User-Item Splitting* also introduce contextual information through a Pre-filtering strategy (Figure 3.6).

Item Splitting

User	Item	Rating
U1	M11	3
U1	M12	5
U1	M11	?

User Splitting

User	Item	Rating
U12	M1	3
U12	M1	5
U11	M1	?

UI Splitting

User	Item	Rating
U12	M11	3
U12	M12	5
U11	M11	?

Figure 3.6 *Example of User Item Splitting approach*

Karatzoglou et al. (Karatzoglou *et al.*, 2010) expand Model-Based Approaches based on Machine Learning (R. Wang *et al.*, 2019; Zhao *et al.*, 2019) to the tensor use. In this way, it is obtained a strategy based on contextual information modelling. The used decomposition is HOSVD (De Lathauwer *et al.*, 2000), exploiting the SVD to multi-dimensional matrices.

Given n users, m items and c contexts and defined the function $l: (f, y) \in R \times Y \mapsto l(f, y) \in R$, estimating the error between the known data Y and the forecast data, Algorithm 1 (Karatzoglou *et al.*, 2010) is used to generate the contextual rating forecasts.

Algorithm 1 *Tensor Factorization*

Input: Y tensor of known ratings, $d \in R$

1. Initializing of $U \in R^{n \times d_U}$, $M \in R^{m \times d_M}$, $C \in R^{c \times d_C}$
2. Initializing of $S \in R^{d_U \times d_M times d_C}$
3. Fixing of $t = t_0$
4. **while** (i, j, k) indices of tensor Y **do**
5. $\eta \leftarrow \frac{1}{\sqrt{t}}$
6. $t \leftarrow t + 1$
7. $F_{ijk} = S \times_U U_{i^\star} \times_M M_{j^\star} \times_C C_{k^\star}$
8. $U_{i^\star} \leftarrow U_{i^\star} - \eta\lambda_U U_{i^\star} - \eta\partial_{U_{i^\star}} l(F_{ijk}, Y_{ijk})$
9. $M_{j^\star} \leftarrow M_{j^\star} - \eta\lambda_M M_{j^\star} - \eta\partial_{M_{j^\star}} l(F_{ijk}, Y_{ijk})$
10. $C_{k^\star} \leftarrow C_{k^\star} - \eta\lambda_C C_{k^\star} - \eta\partial_{C_{k^\star}} l(F_{ijk}, Y_{ijk})$
11. $S \leftarrow S - \eta\lambda_S S - \eta\partial_S l(F_{ijk}, Y_{ijk})$

Output: U, M, C, S

The constants $\lambda_U, \lambda_M, \lambda_C, \lambda_S$ have been fixed by numerical experiments and $U_{i^\star}, M_{j^\star}, C_{k^\star}$ represent the columns of the homonymous matrices evaluated for the current iteration indices.

Another Context Modelling approach for Model-Based Collaborative Filtering is provided by Liu et al. (Liu *et al.*, 2013). This procedure extends the Model-Based Techniques based on Probabilistic Matrix Factorization (Bokde *et al.*, 2015) through the context integration of users and items' latent factors. In this way, two strategies are born:

- *Bayesian Probabilistic Matrix Factorization with Social Relations (BPMFSR)*: the calculation of latent factors associated with users are integrated with personalised hyperparameters (unlike the matrix case) and the parameters associated with social context information are inserted.

- *Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents (BPMFSRIC)*: such strategy is based on BPMFSR on users and a further integration on the calculation of the latent factors associated with the personalised hyperparameter items and the association of the latter based on the social context information (tags) and the specific properties.

The last example of Model-Based Collaborative Filtering with Context Modelling integration is presented by Koren (Koren, 2010). To integrate the temporal context into the Recommender System, Koren develops various analysis models. Be b_{ui} defined, as the forecast about the user u and the item i, the models developed are the following:

- *Static*:

$$b_{ui}(t) = \mu + b_u + b_i \tag{3.8}$$

  where μ is the average of all known ratings about items, b_u is the bias of user and b_i is the bias of the item.

- *Mov*:

$$b_{ui}(t) = \mu + b_u + b_i + b_{i,Bin(t)} \tag{3.9}$$

  where $b_{i,Bin(t)}$ is the bias of item on time interval $Bin(t)$.

- *Linear*:

$$b_{ui}(t) = \mu + b_u + \alpha_u \mathrm{dev}_u(t) + b_i + b_{i,Bin(t)} \tag{3.10}$$

  The undefined elements of the sum are:

  - $dev_u = \mathrm{sign}(t - t_0)|t - t_0|^{\beta}$ where β is a constant empirically determined.
  - α_u is the coefficient associated with the user u.

- *Spline*:

$$b_{ui}(t) = \mu + b_u + \frac{\sum_{l=1}^{k_n} \exp\{-\gamma|t - t_l^n|\} b_{t_l}^n}{\sum_{l=1}^{k_n} \exp\{-\gamma|t - t_l^n|\}} + b_i + b_{i,Bin(t)} \tag{3.11}$$

  where k_n is the number of control time points and γ is a constant empirically evaluated.

- *Linear with daily effect on user bias*:

$$b_{ui}(t) = \mu + b_u + \alpha_u \mathrm{dev}_u(t) + b_{u,t} + b_i + b_{i,Bin(t)} \tag{3.12}$$

where $b_{u,t}$ is the user u bias and has daily variability.

- *Spline with daily effect on user bias:*

$$b_{ui}(t) = \mu + b_u + \frac{\sum_{l=1}^{k_n} \exp\{-\gamma|t - t_l^n|\} b_{t_l}^n}{\sum_{l=1}^{k_n} \exp\{-\gamma|t - t_l^n|\}} + b_{u,t} + b_i + b_{i,Bin(t)} \tag{3.13}$$

Defined the set K of triads (u, i, t) for which the rating $r_{ui}(t)$ is known and chosen the model, context-aware strategy proceeds to minimize the error function:

$$\min \sum_{(u,i,t)\in K} \left(r_{ui}(t) - b_{ui}(t)\right)^2 + \lambda \left(\left\| b_{ui}^{(t)} \right\|_E\right)^2 \tag{3.14}$$

A case of Memory-Based Collaborative Filtering with Post-filtering integration of contextual information is presented in Xu et al. (Xu *et al.*, 2015). This work aims to build a Context-Aware Recommender System for tourist purposes through geolocation of user photos. Both the geographical coordinates of a given location l, $v = (l, u, t)$ the visit associated with location l, user u, time t and the topic based context-aware query $Q = (u_p, s, w, d)$ associated with the target user u_p, the season s, the weather w and the target city d relative to u_p which returns in output a list of places associated with the city d are given. Assuming a set of geo-referenced photos are provided as input.

$$\{P_u : P_u \text{ list of user } u \text{ geo-tagged photos}\}$$

the technique is developed as follows:

1. Construction of tourist locations profiles, containing contextual information

2. Construction of the location database $LDB = \{l_1, \ldots, l_n\}$ with $l_i = \{v_{l_i}, pop(s), pop(w)\}\ \forall i = 1, \ldots, n$

   The elements that make l_i are the visits v_{l_i} associated with the locations l_i, the contextual information $pop(s)$ about the most popular season for location l_i, the contextual information $pop(w)$ about the most popular weather conditions of the location l_i.

3. Construction of the user-location matrix $M = (M_{ul})$ such that M_{ul} indicates the number of times that the user u visited the location l

4. Through the information of the user-location matrix M, the travel history of each user is built.

5. Calculation of the similarity between users

6. Calculation of rating forecasts through

    a) Recovery of N users closest to the specific user

    b) Calculation of the rating forecast for each location

    c) Filtering information obtained through contextual information

    d) Construction of list of m locations with best rating forecasts

3.2.2.1 Context-Aware Matrix Factorization (CAMF)

The Context-Aware Matrix Factorization (CAMF) aims to extend matrix factorization models by analysing bias (*baselines* for contextual conditions) determined by contextual parameters (Baltrunas, Ludwig, *et al.*, 2011).

The calculation of the contextual rating forecast takes place after having derived through a rating matrix factorization technique the $p_i, q_j \in R^d$ vectors associated with the i-th user and j-th item where d is the considered number of latent factors.

It is possible to calculate the forecast $r_{\widehat{ijc_1 \ldots c_k}}$ of the k dimensional nodes associated with evaluated contexts as follows:

$$r_{\widehat{ijc_1 \ldots c_k}} = \overline{r_j} + p_i \cdot q_j + b_i + \sum_{z=1}^{k} B_{jzc_z} \tag{3.15}$$

where $\overline{r_j}$ is the average of ratings of item j, b_i is the baseline parameter for user i, B_{jzc_z} is the parameter that represents the interaction between item j and contextual condition c_z.

Three techniques based on the same idea are presented by Baltrunas et al.:

- ***CAMF-C***: a single parameter is analyzed for each contextual condition which generates bias on ratings. In this method, we have $k = 1$, then we have an only B_{jc_1}.

- ***CAMF-CI***: in the analysis of deviations, items are also considered to determine what effect they have. Such an approach provides the increase in the number of parameters to be analysed, providing for the rating B_{jzc_z} for each item and contextual condition.

- ***CAMF-CC***: items are placed in categories in order to reduce the number of parameters of the CAMF-CI approach losing only a part of the specialization obtained on the calculated deviation. Items of the same category have the same parameter B_{jzc_z} in common.

The calculation of the training values is made by using the Stochastic Gradient Descent on the learning procedure.

$$\min_{p_*,q_*,b_*,B_*} \sum_{r\in R} \left[\left(r_{ijc_1\dots c_k} - \widehat{r_{ijc_1\dots c_k}} \right)^2 + \lambda \left(b_i^2 + \|p_i{}^2\| q_j^2 + \sum_{z=1}^{k} \sum_{c_z=1}^{v_z} B_{jzc_z}^2 \right) \right] \quad (3.16)$$

where $c_z = 1, \dots, v_z$ or $c_z = 0$ if it is unknown, $z = 1, \dots, k$.

Table 3.2 shows a summary of Collaborative Approaches presented above.

Table 3.2 *CARS based on Collaborative Filtering: summary table of some related works*

	Collaborative CARS			
	Pre-filtering	**Post-filtering**	**Context Modelling**	***Contextual information***
	Model-Based			
Baltrunas and Ricci, 2014	X			*Social; Time*
Said, De Luca and Albayrak, 2011	X			*Location*
Zheng, Burke and Mobasher, 2013	X			*Location; Time*
Karatzoglou *et al.*, 2010			X	*Human; Social; Time*
Liu, Wu and Liu, 2013			X	*Social*
Koren, 2010			X	*Time*
Baltrunas, Ludwig and Ricci, 2011			X	*Depends on the dataset*
	Memory-Based			
Xu, Chen and Chen, 2015		X		*Location; Time*

In the next chapter, some challenging applications in the field of Cultural Heritage are discussed, where Context-Aware Recommender Systems are fundamental means to make more addictive the experience of visiting a point of interest: a museum or an archaeological park. In this way, it will be possible to suggest the path more suitable based on a single user's preferences and the context analysis, allowing a more significant interaction of the user with the selected points of interest.

Chapter 4
A Case Study: e-Tourism and Cultural Heritage

Context-Aware Systems have found wide use in various fields, from home automation (domotics) to Artificial Intelligence (AI). One of the areas of more significant application is linked to e-Tourism and, in particular, to Cultural Heritage. The first applications of Context-Aware Computing can be traced back to the beginning of the 1990s, when the adaptive hypermedia systems aimed at museum visits (AlFresco (Stock, 1993), ILEX (Oberlander *et al.*, 1998)) and, in general, to tourism (AVANTI (Fink *et al.*, 1998)) as possible fields of application. AlFresco provided personalised contents as part of a man/machine dialogue around Italian art and combined natural language processing with a hypermedia system connected to a videodisc. Adequately managing the content to be shown to visitors was also the goal of ILEX, which automatically generated hypertext pages with text and images taken from the material collected from existing catalogues and transcripts of conversations with the museum curator. An evolution of the AlFresco system, AVANTI, was a desktop application running on kiosks. It was the first to explore the creation of personalised Web pages of city tours, taking into account the user's interest, knowledge and physical abilities.

4.1 Cultural Heritage and Context Awareness

Cultural Heritage is one of the most critical resources of the territory. It represents one of the scenarios where new technologies can provide more exciting contributions: adaptive systems and related services may increase Cultural Heritage promotion. In fact, tourists can use several services to filter the vast amount of data present on the Network in order to provide only relevant information.

In particular, the cultural tourism market is evolving towards a dimension of complete satisfaction of the tourist's needs. There is an attempt to emphasise, on the one hand, the centrality of the cultural aspect within a 360° travel experience and, on the other, the attention in choosing the "holiday path", accompanied with all the necessary services (transport, accommodation, etc.) according to the Context. The tourist also shows a growing need to play an active and participatory role in the tourist experience, integrating the cultural contents about the visit with unique self-generated contents and sharing them with the "community".

The viral distribution of information, the radical changes in the traveller's decision-making process and the expansion of the knowledge tools available to all connected users are now more than ever the main levers of change. In this regard, the application of Context-Aware Technologies allows offering services at the base of e-Tourism, supporting users, public institutions and sector operators through the automatic, adaptive and real-time recommendation of "core" and "ancillary" services for tourism promotion (D’aniello *et al.*, 2017).

Italy, for example, has a Cultural Heritage that often fails to be fully exploited. The natural, artistic and cultural resources present in many cities, especially in the smallest ones, often remain hidden from tourists (Colace *et al.*, 2016). This problem becomes even more relevant when the tourist has little time to visit a city. For example, think of some passengers on a cruise that they must visit a place unknown to them in a few hours. The problem also arises for those people who, for work, live a temporary experience in a city.

Where to eat? What to see? How to travel to it? These are the typical questions that a user of this kind arises when he is in a station, airport or port. In big cities, there are often pre-established itineraries that tourists can easily find. However, this is not always true in small or medium-sized cities that, even having an important Cultural Heritage, often risk not fully exploiting it.

In this scenario, a Context-Aware System is able to recommend contents and services according to tourists profiles and their context.

As mentioned above, in fact, Context-Aware Computing is conceived for diminishing the cognitive load of users that perform tasks such as retrieving information or accessing services and a wide range of applications is available, with emphasis on Cultural Heritage.

This work explores the possibility of using technologies and methodologies to control the evolution and presentation of information to the user based on different types of Context.

An example of Context Dimension Tree for Tourism domain is shown in Figure 4.1: (a) a specification of some possible Contexts of use for a tourist application; (b) an example of Context instance detected during the execution of a Context-Aware System.

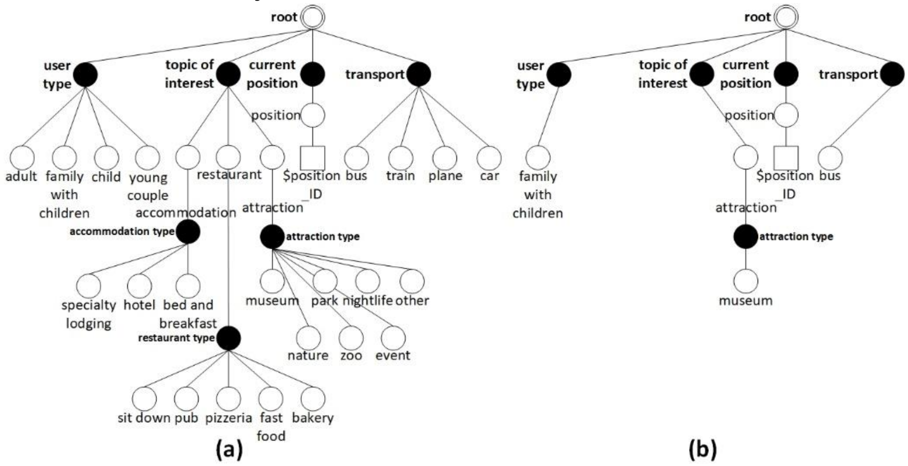

Figure 4.1 *Example of CDT for Tourism*

4.1.1 Digital Storytelling and Context-Awareness for Cultural Heritage Enhancing

«By dint of telling his stories, a man becomes the stories. They continue to live after him, and so he becomes immortal» (from "Big Fish" film). The mystery contained in each 'word' is still one of the most debated issues in humanity's entire history. From this mystery, it comes the narration and its ability to persuade, convert, enchant (Barthes and Duisit, 1975; Robin, 2008).

With the term Storytelling, we mean a learning methodology that consists in 'telling' a story to catch the attention of an audience, convey the message that the story wants to tell and stimulate a specific desire in the readers, transmitting events in words, images and sounds (Alexander and Levine, 2008). We talk about Digital Storytelling, of which there are several definitions, but all turn around the "idea to combine the art of telling stories with the variety of digital multimedia services, such as images, audios and videos" (Robin, 2006): a mixture that allows expressing and narrating, in a vivid way, experiences, situations and considerations (Lambert, 2007).

Digital storytelling exists in many forms and encompasses multiple fields; in particular, there are the following typologies: *linear, non-linear, adaptive, social/collaborative* and *mobile*. Linear storytelling is a traditional narrative form where the author chooses a predefined plot with a fixed sequence of events that the protagonist cannot modify in any way. 'Linear' indicates that any part of the content is seen or listened to in the same order all the times it is enjoyed, confirming the fact that the reader is pushed to follow the narrative path without the possibility to influence the plot and its end (Spaniol *et al.*, 2006). Non-linear storytelling represents a 'not linear' narration form, that is any corpus of contents is structured so that all the possible ways to cover are multiple and variable (Pearce, 1994). The main idea is that the plot is not fixed a priori, but the development of the story and its realization depend on the interaction and the choices of the audience, who from simple reader or spectator becomes a 'co-author'. Instead, the adaptive approach is based on an interactive narration that allows intervening and interacting with the process of construction of the story adapting it to the alterations caused by the intervention of the user, with the purpose to keep a particular narrative coherence (Kickmeier-Rust *et al.*, 2008). The use of media and social media to share contents and narrative artefacts has considerably transformed even how the stories can be described and told (Lee *et al.*, 2014). 'Collaborative' and 'social' storytelling are forms of collective construction of a story where the shared narrations become socially interactive means that support and foments the users continuous co-participations (Gabriel and Connell, 2010).

If technology has changed how to tell the stories, new mobile tools have solved the overlapping of the narratological and the spatial parts (Ye and Papavassiliou, 2008; Napoletano *et al.*, 2015). Consequently, the researchers and the developers have been pushed to define new approaches able to integrate the two dimensions in order to make the cultural experience as immediately as possible and at the same time ubiquitous. Also, the support of geolocation (positioning and localization through integrated GPS) has further improved the use of these mobile tools, allowing a solid reinforcement of the visitor's experience (Tiwari and Kaushik, 2017). In fact, it is very current the theme of the creation of geolocated interactive narrations aware of the context, that is, of stories whose contents, organized in a narrative sequence, are firmly anchored to the physical places and other significant parameters of the environmental context (Pittarello, 2006; Colace, De Santo, Greco, Lemma, *et al.*, 2014; Chang *et al.*, 2015). Therefore, it is understandable that it is essential to give the tourist adaptive services and contents able to transmit the correct information in the right contexts (Francesco Colace, Moscato, *et al.*, 2015).

A fundamental contribution to this scenario has come from the increasing diffusion of smart mobile devices. The latter led to a concrete use of Pervasive and Context-Aware Computing approaches (Francesco Colace, De Santo, Moscato, *et al.*, 2015a, 2015b) to distribute contents and services (Francesco Colace, De Santo, Greco, *et al.*, 2015).

In this respect, there are other several cases of use of Digital Storytelling, in order to make the visitors' experience more interactive and engaging, proposing personalised stories through new technologies (not only geolocation and mobile devices but also social networks, augmented reality, gaming, etc.).

In (El-Nasr, 2007), Young has proposed an interactive narrative architecture that "re-projects" the story's events, using the planning. In the wake of the work by Young, many adaptive narrative theories have been developed, such as the use of an algorithm of user modelling to predict his/her actions and the trigger events in real-time; in this way, the character of the user is given by his/her actions with respect to the context of the story.

It is then possible to identify and list a very long series of 'collaborative' and 'social' storytelling with purposes of pre-schooling, amusement, teaching and fruition of Cultural Heritage. Among these, we would like to remember Casting (Schumann, Buttler and Lukosch, 2010): a software system that supports the audio-based narration of collaborative groups to create non-linear stories.

Moreover, the user's experience, thanks to the mobility and the territorial and thematic localization, can renew itself using the physical environment like background and plot on which basis structuring and defining stories that wind along with the urban topography (Paay et al., 2008).

The use of augmented reality is one of the services able to improve in a substantial way the interaction and the active participation of the visitor, bringing him/her in a more engaging way nearer to the narration of the place, superimposing the actual reality to a virtual one that stimulates the imagination and favours sense connections (Bertino et al., 2005).

Frequency 1550 (Yiannoutsou and Avouris, 2012) is a classic example of role-playing in augmented reality, presented by Waag Society and realized to allow the students to get closer to the medieval history of the city of Amsterdam. Thanks to smartphone devices, the visitors move in a hybrid reality that alternates the modern and current city to the faraway one of 1550. Structured as a treasure hunt, the students move inside the city, guided by the mobile device, searching for the solution of the story.

As shown in Chapter 5, the proposed research objective also seeks to contribute to this respect: to identify an approach for realising a dynamic storytelling engine that can allow the dynamic supply of narrative contents, not necessarily predetermined and pertinent to the needs and the behaviours of the users. The adaptability to the context (1), the social networks (2) and the mobile world (3) represent the pivotal points in order to give a technological solution to digital storytelling which combines, indeed, the adaptive, social and mobile approaches.

1. The adaptive approach enhances the idea of a narrative experience able to change according to users and their context dynamically.

2. The social component's fundamental feature is the sharing of the author's role among many individuals, each of which, in turn, takes part in creating a portion of the story and none of them can assume its paternity.

3. A mobile application is a perfect tool for all necessary information in real-time, facilitating access to various helpful information.

4.2 CARS in Cultural Heritage

In order to identify content and services suitable for the needs of large masses of tourists, it is possible to define appropriate models and solutions of fruition that make the visitor experience more appealing and immersive based on the current context.

In particular, the introduction of a Context-Aware Recommender System (CARS) is useful to achieve this goal: an approach for the recommendation of information depending on the users and their context, through techniques of dynamic profiling and solutions for the generation of content and services coherent with the experiential mission of the visit.

In fact, based on the social behaviours of each user, of his/her interests, and contextual information, it is possible to create a model in order to elaborate the obtained data and propose a personalised *story* to users, including a series of information about the place where they are and about other places they can visit.

The Context Evolution System (CES), presented in (Chianese and Piccialli, 2016), provides an initial model for applying CARS within the field of Cultural Heritage. Specifically, an application introduced for an exhibition called "the Beauty of the Truth" in Naples is presented as a case study.

The main elements of this approach are:

- *Service Engine*: the main component of architecture and consists of three modules.
    - *Events Detector* aims at event recognition and activation of the *Context Switching Computation Module*.
    - *Context Switching Computation Module* aims to identify possible changes in context and select Contextual Data Views (data for the specific situation) and Basket of Services (services adapted to the contextual situation).
    - *Visiting Paths Generator* aims to determine a path to the user when visiting items.

- *Services Deliverer*: the component dedicated to the analysis of the status of user-items systems. After analysing the nearby items through the *Events Detector* and determining the user context through the Context Switching Computation Module, the Services Deliverer allows the transmission of information adapted to the user through a multimedia guide.
- *Knowledge Base* and *User LOG*: the data management elements to propose appropriated items to the user's preferences and behaviour.
- *Context Manager*: the component suitable for the analysis of information to determine the contextual state.

This architecture fits into an environment capable of assessing specific contextual situations through a set of factors underlying Smart Cities (Schaffers *et al.*, 2011). In particular, the concept of Single Smart Space (S^3) is developed (Chianese and Piccialli, 2016).

The fundamental phase of the analysis of the contextual situation in S^3 is the study of *Context Evolution Graph*, $CEG = (C;\ \Sigma; l_c; l_s)$, consisting of the following components:

- $C = \{c_1, \dots, c_n\}$, the set of contextual variables;
- $\Sigma \subseteq C \times C$, the set of graph edges;
- $l_c: c \in C \mapsto (v, S)$, functions that link each contextual variable with the Contextual Data $v \in V$ and the Basket of Services S;
- $l_s: t \in \Sigma \mapsto e \in E$, functions that link each edge of the graph with the specific event e.

Another interesting approach to visiting a museum is provided by SMARTMUSEUM (Ruotsalo *et al.*, 2013). This application is based on a Content-Based Approach with the introduction of contextual information through a Pre-filtering Technique.

The architecture presented is composed of four main components:

- *Metadata Service*, which stores the data obtained through the Web;
- *Context Service*,
- *User Profile Service*, which generates the user profile through contextual information and the use of ontologies. In addition, known feedback, which is relevant in the contextual context, is also used through a probabilistic approach;
- *Filtering Service*, which indexes the items through the Web information and, then, carries out the recommendation process.

(Bartolini *et al.*, 2016) is instead an alternative work involving a Content-Based Recommendation Technique with dual filtering phase of the context: Pre/Post-Filtering Strategy.

The CARS model is evaluated in two case studies: an outdoor case related to the archaeological site of Paestum and an indoor area consisting of the Capodimonte National Museum.

The main components are:

- *Multimedia Data Management Engine*: the component that manages numerous functionalities and can be considered the heart of the architecture. Specifically, this component:
  - accesses the *Indexing and Access Manager Module* containing the contents associated with the items;
  - through the *Feature Extraction Module*, captures multimedia data. These are used for indexing items and for obtaining a data structural description data. Finally, the data is stored in the *Multimedia Storage and Staging*;
- *Sensor Management Middleware*: an element able to interact with sensors in a specific area to determine contextual information. These are stored in the *Knowledge Base*;
- *Knowledge Base*: the component that manages geographical information about user location and stores contextual information provided by the *Sensor Management Middleware*. It also contains information about user preferences and descriptions about points of interest;
- *Multimedia Recommender Engine* that, through various components, constitutes the *Recommendation Module*. It is composed of:
  - *Candidate Set Building Module*, that selects suitable items to recommend to the user;
  - *Object Ranking Module*, that generates ratings for the elements selected by the *Candidate Set Building Module*;
  - *Visiting Paths Generator*, that dynamically selects a subset of items based on contextual information and user actions and proposes a visit path about the site or the museum in question.

The Pre-filtering Phase consists of the selection of items in the *Candidate Set Building Module*. These are chosen according to the user location and preferences or needs. The Post-filtering phase is carried out based on contextual information not considered in the Pre-filtering Phase and aimed at constructing the visit path.

The Recommender System consisting of the *Object Ranking Module* is a reworking of the recommendation mode associated with the PageRank (Page *et al.*, 1998; Albanese *et al.*, 2011).

Finally, Turist@ (Batet *et al.*, 2012) presents an architecture based on a multi-agent structure that can be classified as follows:

- *User Agent*: this is the component that allows the user to interact with the application and its functionality. It also allows building a vector of preferences for the specific user sent to the Recommender Agent through some preliminary questions. Finally, through the *Agent*, the user can filter the activities and request personalised suggestions;
- *Activity Agents*: they are *Agents* associated with single activities. Moreover, they are related to a specific database for the activity to which they are dedicated;
- *Broken Agent*: it aims to connect the *User Agent* with the *Activity Agents* in order to make communication more efficient and reduce the time needed to select the correct information;
- *Recommender System*: the name suggests that it is the component for calculating recommendations. The user profile, initially provided by the *User Agent*, is modified dynamically based on the interaction between the user and the system. It also contains a database of the main features of some items to provide quick initial suggestions.

The most exciting features of this application are the ability to dynamically update the user profile and the double choice to make the recommendation (Content-Based/Collaborative Filtering). The suggestions consider the user's position, acquired through the mobile device's GPS, and the activities to be considered, whose location is stored in a specific database.

Context-Aware Recommender Systems have been described and classified, and several approaches for introducing contextual information within Recommender Systems have been shown. The variety and heterogeneity of these approaches testify to the significant number of studies about this field and the wide diffusion that CARS have had in recent years.

In particular, Recommender Systems that exploit contextual information represent crucial tools for improving users' experience in the Cultural Heritage field. Through CARS, archaeological sites and museums are made interactive and unique.

Future challenges concern new complex scenarios in which a change of context causes a transformation of the experience that is about to be lived.

Chapter 5
The Proposed System

The present work aims at the development of an innovative Recommender System, based on Context-Aware Computing, to support citizens and, in particular, tourists, to provide them with advanced services that, highly customizable, able to allow, through the usage of the new technologies, more involving, challenging and appealing fruition of information compared to the current patterns.

For example, the intention is to improve a tourist's experience and quickly lead him/her to the new context where he/she lives. It is indeed crucial for a person, who is in a place to visit, to have the possibility to get oriented among the points of interest that the place offers and to get to know its history, in order to be aware of the available cultural attractions. As soon as he/she arrives at a new location, a tourist will need to know, in general terms, the characteristics of the place and why it is worth a visit. Through the provided services, he/she will immediately obtain a description to deepen all the details at a later stage.

Another classic situation is when the tourist finds himself planning his/her activities and his/her visits. In that case, using the system which it intends to implement, the tourist will be recommended dynamic itineraries based on his/her tastes and of the parameters related to the current context: during the planning of the itinerary, the system has to always have in evidence and adapt to dashboards of measurement of the resources that the consumer has made available (such as time, budget, number of members of the group and age).

In this scenario, the system might be translated into mobile applications of "trip designer" type that build a travel itinerary drawing from default folders the various steps of which the itinerary itself is composed, or by means of a chatbot, that maintains, through natural language processing and context recognition techniques, a logical talk with the user in order to meet specific touristic needs. The system will not have to limit itself to provide a simple list of all the data found about that place, but it shall present such data in a way that the consumer can be active and look into the past, the present and the future of it, behaving like a modern touristic guide, also using social networks, that are by now an integral part of our daily lives and containers of vast amounts of information.

The system will be able to detect the whole context and react to it, providing the consumer with suitable answers in terms of services. For example, let us imagine a lousy weather forecast for the whole trip: in that case, the system can discard the outdoor activities and suggest only the "indoor services".

For this purpose, the first steps will be to present a system that shows a high level of Context Awareness. It should be understood as a set of technical features capable of providing added value to services in different touristic/cultural scenarios.

5.1 Proposed Architecture

As shown in Figure 5.1, the proposed system architecture is composed of different blocks that perform the following main functions:

- Data gathering and services establishment
- Context rendering and use of said context for the recommendation of contents and services
- Presentation of selected contents/services

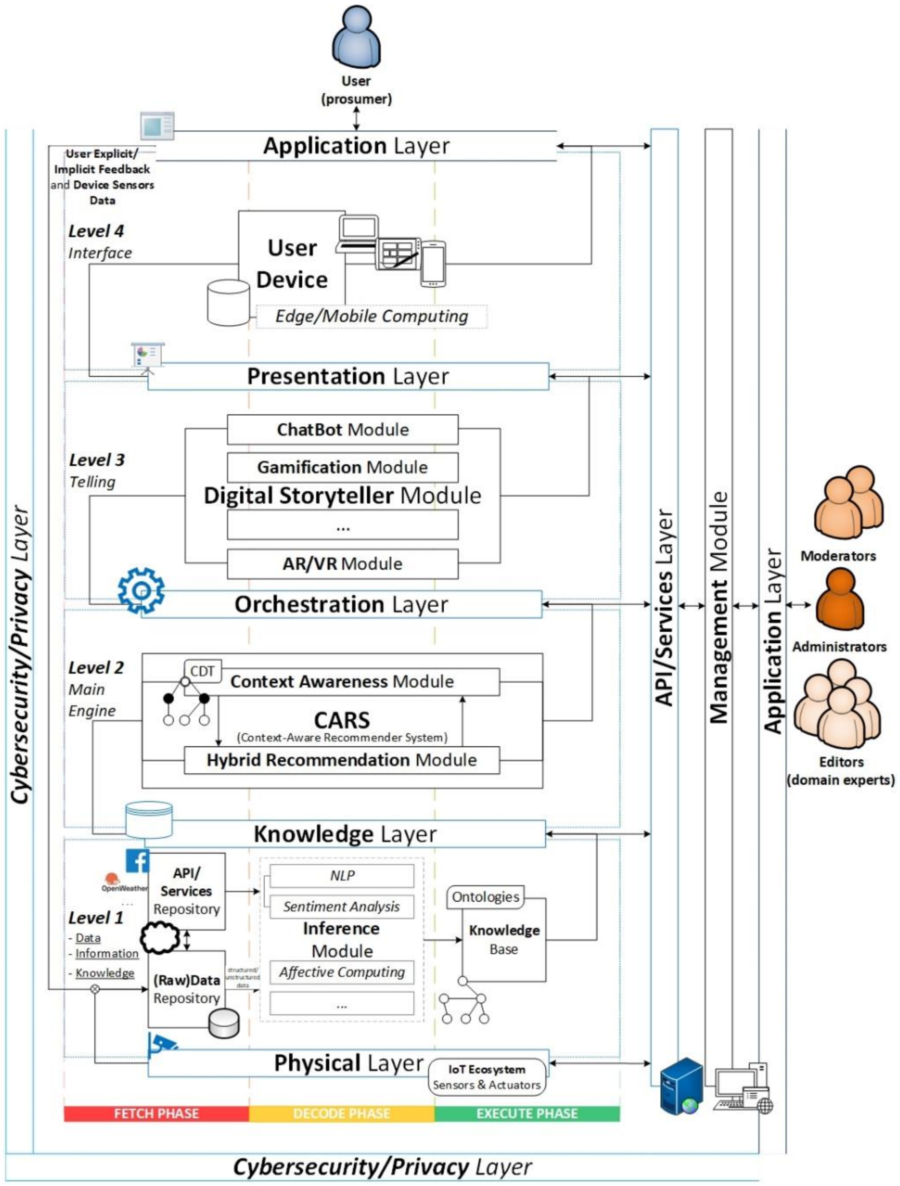

Figure 5.1 *The proposed system architecture*

A. **Physical** Layer

It mainly consists of sensors that acquire information for the system and actuators that respond to commands from the system.

1. **Level 1.** Inference Engines: from data to knowledge

The system, thought to be continuously functioning, collects data from various sources without interruption and immediately processes them, depending on the users and the events. The latter, detected and analysed, will have to be translated into facts associated with specific semantic values: it is, therefore, necessary to use inferential engines capable of drawing conclusions by applying to reported facts particular rules.

For example, an approach to implement inferential engines comes from the so-called "Bayesian networks": powerful conceptual, mathematic and application tools allowing to manage the complex problems with a significant number of variables interlinked by means of both probabilistic and deterministic relations. Such networks also allow for updating the probabilities of all the variables at stake, any time that new information on some of them are collected, by using the Bayes theorem.

From another point of view, the proposed approach envisages functional blocks with three main phases of functioning:

I. Collection phase (*fetch phase*): data, referred to as "rough data", are provided by different types of sources (for example, physical sensors or API services). The set of data that are most significant with a view to the analysis that is meant to be carried out is saved within a database repository.

II. Pre-elaboration phase (*decode phase*): in this phase, data are transformed in order to adapt them to the system that will have to use them. In general, data come indeed from different sources and therefore show inconsistencies such as, for instance, the usage of different denominations to identify the same value of a feature. In addition, this phase envisions cleaning the collected data in order to eliminate any error and the treatment of missing data. Such a phase ends up with sampling and making such data discrete.
This phase may also include the use of advanced techniques such as Sentiment Analysis or Affective Computing, which leverage human-computer interaction, information retrieval and multimodal signal processing for distilling people's sentiments from the ever-growing amount of online social data (Cambria, 2016).

III. Elaboration phase (*execute phase*): this phase aims at providing a representation and interpretation of the acquired knowledge, starting from information correctly memorised. To this end, an approach is followed based on the main views previously described, leading to implementing and using "decisional models". Such models are continuously improved based on newly collected data and experiences or previously treated cases.

B. **Knowledge** Layer

In such a scenario, data represent the key to enabling the implementation of vertical services: the goal is to implement a Knowledge Base (KB) to collect, elaborate, and managing information in real-time. In this respect, by Knowledge Organization System (KOS), we mean well-known schemes such as taxonomies, thesauri and other types of vocabularies that, together with ontologies, constitute valuable tools to shape the reality of interest into concepts and relations between concepts.

Many benefits stem from this: using ontologies, for instance, allows to fix a series of key concepts and definitions relating to a given domain that can be shared, thus making the appropriate terminologies available (collaborative knowledge sharing); furthermore, an ontology allows a full re-usage of the knowledge that it codifies, even within other ontologies or instead for their completion (non-redundancy of information) and, being susceptible to interpretation by electronic calculators, enables the automatic treatment of knowledge with relevant significant advantages (Semantic Web).

2. **Level 2.** Main Engine: contextual recommendations

By defining a series of services beneficial to the tourist (basket of services), it is necessary to integrate the context to the recommendation process to provide valid suggestions under certain conditions. That is the purpose of the recommendation engine. For example, if it is raining and the user is walking, he would prefer to receive recommendations of interest sites that are not too far away.

A deep understanding of the context problem is essential. For this reason, the main contribution in this phase is the modelling of context through graphic formalism, like the Context Dimension Tree (CDT). This tree-shaped model, along with the use of the field ontologies, can both represent the context in its main dimensions and interrogate each reference database to select the right services/data efficiently.

Based on this type of representation, the proposed methodology is divided into three main phases: the phase of design of the Context Tree, where it is possible to identify relevant context elements for the application considered; the phase of definition of partial views, where it is possible to match each one to a different set of services/data; the phase of composition of global views, where it is possible to elaborate the final responses to the queries.

Thus, it is possible, for example, to create a custom itinerary of the visit according to the context and the user through the selection of appropriate services for:

- ✓ Planning the itinerary of a trip, customizable in a few steps, and facilitating transportation; during this phase, the client can taste the different steps about the itinerary, obtaining detailed information on the available activities.
- ✓ Organizing visits to museums or art location, preferably without having to queue and with custom discounts.
- ✓ Dynamically re-planning the itinerary based on context and user behaviour or emergencies (automatic "travel assistant" services that can suggest actions and modifications to the itinerary).
- ✓ Promoting the discovery of unusual places marked by incredible beauty, high cultural value and enogastronomic heritage, the primary example of Made in Italy excellence.

More details on the functioning of the Context-Aware Recommender System, in terms of its main modules (Context Awareness Module and Hybrid Recommendation Module), are explained in the following section.

C. **Orchestration** Layer

The orchestration layer uses context and mash-up techniques capable of recovering information from different sources and overcoming content limitations and "pre-packaged" services, whose structures must, instead, vary based on the situation during which the fruition happens. The main objective is to integrate data and services through a single interface, having the correct information according to user and context needs.

An example of the benefits is the integration of primary content and services to support services for the tourist (*Consumer Mashup*), like maps or weather info, and/or business services through API (*Business Mashup*), like booking or purchasing a museum ticket, allowing to upgrade the user experience and the accessibility of content and services. Another example is using more than one source for the presentation of the "same" information, like data on cultural heritage acquired by physical sensors or through Open Data (*Data Mashup*). The information is combined to obtain a detailed picture of the entity (such as work of art and site of interest) analysed by different tools.

3. **Level 3.** Telling

One of the main goals could be to combine context recognition and digital storytelling techniques to provide, through a custom narrative, the information related to the location. The aim is to make results available like a modern tourist guide that, thanks to a "storytelling engine", can provide dynamically different narrative contents, as services, accordingly integrated, not necessarily predetermined and adherent to the needs and behaviours of the users.

The "digital narration" includes:

- ✓ Information on the location of the visit (main characteristics and historical notions): the story narrated in the first person by the host (memories, autobiography, family traditions) and stories lived or set in the locations where we are located (novels, legends, songs, movies, historical events).
- ✓ Points of interest for the user with all related services, filtered by category and accompanied by multimedia in-depth information.
- ✓ Experiences lived by other users, like authentic testimonies of the destination: the site of cultural interest users could be involved in the making of new digital resources (stories/comments, images and videos) that, stimulated, gathered and framed in the best way possible, will contribute to the enrichment and development of new compelling and personal stories.

This level can make the user experience more immersive and participative. The audience is engaged more interactively and the environment is evolving into a more complex transmedia scenario, integrating different technologies such as Augmented Reality (AR), Virtual Reality (VR) and gaming elements.

D. **Presentation** Layer

The presentation-oriented layer is used to adapt the presentation of content and services according to the users' needs and preferences.

E. **Level 4**. Interface

It mainly depends on the user device type, such as notebook, smartphone, tablet and smartwatch. In general, it refers to Mobile Computing as users' ability to access IT services regardless of their location. This interface enables new functionalities, such as determining and tracking the user's location and acquiring information collaboratively.

F. **Application** Layer

The application-oriented layer adapts content and services according to the different channels and devices (application layer).

The application layer includes all the software necessary to offer a specific service. As a result of this processing process, the data are available to real applications on different devices (smartphones, tablets, and wearable devices).

Each horizontal layer can communicate with another layer through an API/Services Layer. In practice, it abstracts business logic and data access.

Data and operating parameters can be entered and set manually through a Management Module in order to, for example, enable the use of experts preference to enhance the system's performance.

Finally, an important role is played by the layer that includes aspects related to privacy and cybersecurity.

In this first phase, the research activity was focused on the Context-Aware Recommendation Module. In this regard, an approach to recommend contents and services to users based on context was proposed.

5.2 Proposed Approach

The innovative characteristics of the proposed approach mainly have to do with the informational content that is intended to be made available to the end-users, suggesting different points of view, such as:

- Context representation

- Recommendation engine

5.2.1 Context representation

The goal is primarily to deliver to different categories of users, in a given moment, information that is useful in a given context: in practice, the objective would be setting up an architecture characterized by a high degree of Context Awareness. Real-time understanding of the context where users are located, via a representation through graphs, allows indeed to provide them with a wide array of personalised, "tailored" services and hints regarding the decisions to make. It can help them in professional and private daily life, managing in the best possible way both the time and resources they have and showing them what is around, ultimately meeting their needs.

Context Awareness should be understood as a set of technical features capable of providing added value to services in different operational segments. Context-Aware Computing applications can exploit, in this specific case, such features in order to provide users with context-related information or to suggest them an appropriate selection of actions. In order to achieve a better representation of the various features, formal tools of context representation have been adopted, capable of defining in details the user's needs in the context where he is acting, through an approach «who, where, why, when, what, how».

More in detail, the representation of the context will be implemented by means of formal models of representation, such as the Context DimensionTree (CDT).

Figure 5.2 shows a general designed CDT, called Meta-CDT, which is the starting point for the design of a specific CDT that can be exploited in contextual applications. It is possible to notice six top dimension nodes, which correspond to the 5W1H method.

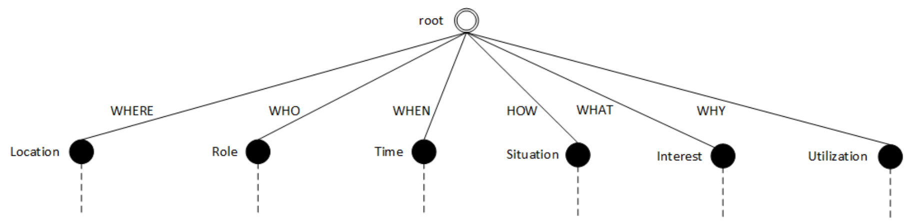

Figure 5.2 *Meta-CDT*

5.2.1.1 The Use of the CDT

The methodology based on the Context Dimension Tree can be seen as composed of three main steps: the step of the design of the contexts' tree (1), the step of the definition of all the partial views (2) and the step of the composition of all the global views (3).

1. In this step, the Context Dimension Tree is used to identify all the relevant elements of the context for the considered application.

2. After defining all the contexts and all their context elements, in this step, a different portion of the database is associated with every context element, containing relevant data for it.

3. This one is the step in which one has the automatic build of all the views associated with every context, which is made starting from the partial views related to the context elements. Once created all the global views of the contexts, all the answers to the questions that will be asked to the system will be processed from these views and, in particular, from the view associated with the context in which one is located when the question is asked.

5.2.2 Recommendation Engine

This work aims to introduce a Context-Aware Recommender System based on Collaborative Filtering and Contextual Modelling.

The proposed approach involves analysing the entire context in which a user may be in a given moment. In other words, the main idea is to consider the contextual information as a group and then develop the bias related to user and item according to the specific context identified.

The prediction of contextualised ratings is improved by analyzing user-item affinity obtained through Matrix Factorization techniques on known decontextualised ratings. This technique is widely used in traditional Recommender Systems based on Collaborative Filtering.

5.2.2.1 The Embedded Context

In the Context-Aware Recommender Systems (CARS) field, several approaches have been developed to integrate context within recommendations based on both Collaborative Filtering (CF) and Content-Based (CB) methods (Adomavicius and Tuzhilin, 2015; Villegas *et al.*, 2018).

Important examples are the Context-Aware Matrix Factorization (CAMF) techniques (Baltrunas, Ludwig, *et al.*, 2011) and the Splitting techniques (Baltrunas and Ricci, 2009, 2014; Said *et al.*, 2011; Zheng *et al.*, 2013).

However, these techniques aim to consider each contextual information individually in order to simplify the calculation of rating forecasts.

Consider the example of Table 5.1.

Table 5.1 *Example of contextual information*

Location Context	**Time Context**
Museum (M)	Weekend (E)
Archeological Park (P)	Weekday (D)

As previously mentioned, CARS would usually evaluate Location Context and Time Context individually.

For example, the CAMF-CI (CAMF Context/Item) method will create parameters associated with the items and values "Museum" and "Archeological Park" for the Location and, separately, parameters related to the items and values "Weekend" and "Weekday" for the Time.

These parameters will correct the not contextualised rating forecast ($\widehat{r_{ui}}$) associated with the user u and the item i, generating the contextualised forecast. Specifically, it will have:

$$\widehat{r_{uic_1c_2}} = \bar{r} + p_u q_i + b_u + B_{i,c_1} + B_{i,c_2} \text{ with } c_1 \in \{\text{M, P}\} \text{ e } c_2 \in \{\text{E, D}\}$$

where $\bar{r}$ represents the average of known ratings, $p_u q_i$ is the element associated with Matrix Factorization, b_u is the user bias and the parameters B_{i,c_1} and B_{i,c_2} integrate contextual information into the analysis of the items.

To give another example, in Splitting techniques (User Splitting or Item Splitting), there is a preference to use single contextual information directly in order to maintain a linear computational complexity. The User/Item Splitting method allows splitting one contextual information on users and one contextual information on items (Table 5.2).

Table 5.2 *Example of User/Item Splitting that uses the contextual information of* Table 5.1

	$i_1 - Weekend$	$i_1 - Weekday$	...
$u_1 - Museum$	$\widehat{r_{u_1,i_1,M,E}}$	$\widehat{r_{u_1,i_1,M,D}}$	...
$u_1 - ArcheologicalPark$	$\widehat{r_{u_1,i_1,P,E}}$	$\widehat{r_{u_1,i_1,P,D}}$	...
...	...	...	$\ddots$

Besides, in the Context-Aware Recommender Systems based on Collaborative Filtering, the representation of known ratings can be done through the Tensors tool (De Lathauwer *et al.*, 2000; Kolda and Bader, 2009). This situation implies that it is possible to represent the single rating precisely in the position associated with all known contextual information, unlike the approaches mentioned above.

In this regard, it is possible to use directly the Tensorial Factorization techniques associated with Machine Learning. These techniques are more precise but may require a high computational cost for rating forecasts.

For example, returning to the information of Table 5.1, a fourth-order tensor will be obtained in which the known ratings are associated simultaneously with both the value assumed by the Location Context and the value assumed by the Time Context.

The present work aims to consider contextual information as a whole, in the same way as tensors, without taking advantage of Tensor Factorization, which considerably reduces the computational cost.

In this way, the Location Context and Time Context will not be analyzed individually, as in the examples discussed above. However, the contextual information pairs will be considered, as shown in Table 5.3.

Table 5.3 *List of Embedded Contexts based on contextual information of* Table 5.1

Embedded Context
(Museum, Weekend)
(Museum, Weekday)
(Archeological Park, Weekend)
(Archeological Park, Weekday)

This approach allows using the available information, making it more consistent with the effects that combinations of contextual information have on users and items, i.e., rating forecasts.

This consideration will be based on the proposed approach: a method based on Matrix Factorization able to directly exploit the entire context that, in this case, it is called Embedded Context.

5.2.2.2 Contextual Bias and Matrix Factorization

Within Model-Based Recommender Systems using Matrix Factoring, rating forecasts can be obtained as follows:

$$\widehat{r_{ui}} = \bar{r} + b_u + b_i + p_u q_i \tag{5.1}$$

where $\bar{r}$ represents the average of all known ratings, b_u, and b_i represent the bias related to the user u and the item i and, finally, $p_u q_i$ is obtained through factorization techniques on the matrix of known ratings.

The proposed approach's main idea is to introduce bias related to the entire context being considered (Embedded Context).

This approach will allow calculating the rating forecast $\widehat{r_{uic}}$ associated with the user u, the item i and the Embedded Context c through the following formula.

$$\widehat{r_{uic}} = \overline{r_c} + b_{uc} + b_{ic} + p_u q_i \tag{5.2}$$

where $\overline{r_c}$ represents the average of the known ratings in the Embedded Context c, b_{uc} and b_{ic} are the bias of the user u and the item i in the Embedded Context c respectively. The item $p_u q_i$ integrates the user/item affinity obtained without considering the contextual information known in calculating the contextual rating forecast.

Thus, given the contextual values of Table 5.1, the elements shown in Table 5.4 will be calculated using the known values.

Table 5.4 *Parameters required to obtain the rating forecasts based on the proposed approach*

Embedded Context	*Average*	*Bias User*	*Bias Item*
$\{Museum, Weekend\}$	$\overline{r_{(M,E)}}$	$b_{u,(M,E)}$	$b_{i,(M,E)}$
$\{Museum, Weekday\}$	$\overline{r_{(M,D)}}$	$b_{u,(M,D)}$	$b_{i,(M,D)}$
$\{Archeological\ Park, Weekend\}$	$\overline{r_{(P,E)}}$	$b_{u,(P,E)}$	$b_{i,(P,E)}$
$\{Archeological\ Park, Weekday\}$	$\overline{r_{(P,D)}}$	$b_{u,(P,D)}$	$b_{i,(P,D)}$

5.2.3 The Cold Start Problem

In a real scenario, to overcome the cold start problem, a hybrid approach has been adopted by introducing a Knowledge-Based Filtering based on social networks to allow new users of the system to interact with it and still get initial contextual recommendations. This approach will be used to fill the rating matrix and the collaborative recommendation approach proposed in the previous paragraph can be used.

In order to determine the user's interests (*user features vector*), as well as the categories to be recommended, its most recent LIKEs on Facebook pages are taken into consideration: each LIKE corresponds to a specific subcategory to be associated with a category of TripAdvisor; for example, to a LIKE on a page relating to an art gallery or a painter will correspond an occurrence of the category "museum". Using the proposed approach, it is possible to calculate the number of occurrences and obtain the preferred categories of interest; then, based on the indexes of recommendation obtained, the user will be offered a personalised list of resources.

To model the system, we have chosen to use an ontological representation: an oriented and labelled graph model to represent Web resources. In particular, the ontology is made up of Classes and Properties: The Classes can be considered as groupings of Individuals sharing the same characteristics; the Classes are intended as binary relations between Individuals (instances of classes) or between Individuals and Values. The basic unit, therefore, is given by the so-called statement, i.e. a triple of the type: $Subject - Predicate - Object$.

Based on what has been said, it has been possible to define the graph shown in Figure 5.3. As can be seen, the ontological model proposed consists of six nodes representing the main classes of the domain of interest:

- $User$ is a consumer of the system (examples of instances of User: Dominic, Frank, Mark, ...);
- $Friend$ is a friend of the user;
- $Interest$ is a type of interest that is associated with a user's LIKE (instances of Interest: school, university, pizza, night club, ...);
- $Category$ represents a set of interests (e.g., cultural, restaurant, nightlife, ...);
- $Place$ is any place, represented by a couple of geographical coordinates, to which are associated TAGs of users (Salerno, University of Salerno, ...);
- $Resource$ is any resource with a specific category that can be proposed to the user (the Greek Temples of Paestum, the restaurant "Gusto Italiano", ...).

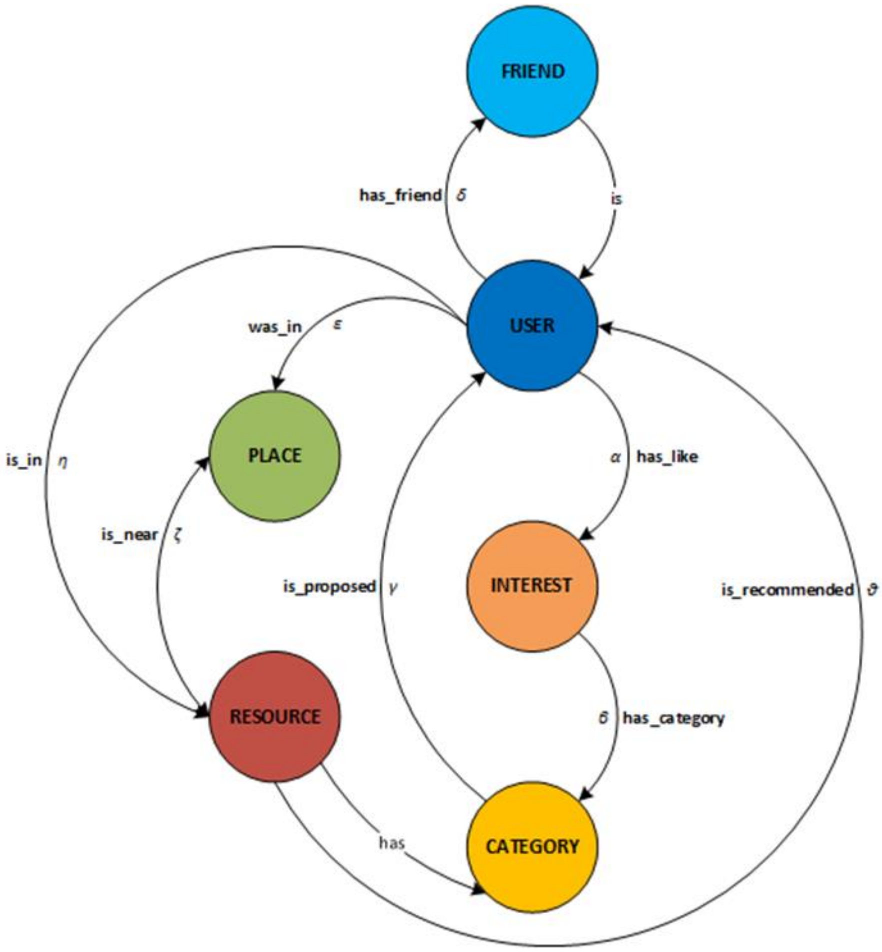

Figure 5.3 *Model used to calculate the recommendation indexes*

Nodes are linked by arcs, which represent properties between classes:

- has_friend: a user can have one or more friends;
- has_like: a user may be interested in one or more types of interest;
- $has\_category$: each type of interest is associated with one or more categories;
- $is\_proposed$: categories are proposed to the user;
- was_in: a user has been in a specific place;
- is_near: a place is near a resource;
- is_in: a user can be in a resource;
- $is\_recommended$: resources are recommended to the user.

Starting from the ontological model consisting of concepts (nodes) and relationships (arcs), a system has been defined by attributing to each semantic relationship a value between 0 and 1 (weighted graph):

- $\alpha = weight_{ui}$ represents the number of LIKE of a user u regarding a specific type of interest i ($\alpha \geq 0, \alpha \in N$);
- $\beta = weight_{ic}$ is the conformity degree between a type of interest i and a category c ($0 \leq \beta \geq 1$);
- $\gamma = weight_{cu}$ represents the recommender degree of a category c and a users u ($\gamma \geq 0$);
- $\delta = weight_{uu}$ represents the degree of friendship between a user u and a friend of the user f ($0 \leq \delta \geq 1, with\ \delta = 1\ if\ u = f$);
- $\varepsilon = weight_{up}$ is the number of the TAG of the user u at the place p ($\varepsilon \geq 0, \varepsilon \in N$);
- $\zeta = weight_{pr}$ is the proximity degree of a place p to a resource r ($0 \leq \zeta \geq 1, with\ \zeta = 1\ if\ p \equiv r$);
- $\eta = weight_{ur}$ represents the proximity of a user u to a resource r ($0 \leq \eta \geq 1, with\ \eta = 1\ if\ u\ is\ in\ r$);
- $\theta = weight_{ru}$ represents the degree of the recommendation of a resource r to a user u ($\theta \geq 0$).

Finally, it was possible to implement a reasoner able to recommend resources to the user based on the categories of interest and identified resources, using two main indexes:

- I_{cat}: index related to the liking of a category in order to calculate the vector of user features, as (Casillo, Colace, *et al.*, 2019);
- I_{res}: index related to the liking of a resource in order to provide contextual recommendations (Casillo, Colace, *et al.*, 2019).

In particular, the rating index for a category (I_{cat}) represents the interest of a user in a specific category. It is calculated considering the number of LIKEs of the user and his friends related to social pages. This index is associated with a type of interest and the index of correspondence of this to the category itself, dividing by all friends, if the user has no interest in that category, or, if not, by the number of friends who have interest in the category considered. Subsequently, the category index is normalised, considering the sum of the indexes of all categories for that particular user.

Particular interest has been placed on the calculation of the degree of friendship or the weight of the relationship between two users. In this regard, not all friends affect the same way and, for this reason, an index of friendship between users should be considered: the more a user is a friend with another, the more the interests of the friend will affect the liking of a category for the user himself. To do this, one needs to know that friendship is not necessarily a reflective relationship, i.e. user A may consider user B as "best friend" while user B considers "friend" user A. In particular, the weight of the friendship between two users can be calculated as a sum of multiplications between the number of each communication parameter (LIKEs, messages, tags) and their weight (Casillo, Colace, *et al.*, 2019).

The recommendation index of a resource (I_{res}) represents, instead, the potential liking of the user towards a specific resource. It is defined by the normalised category index, the proximity of the resource by the user, the rating of the resource and the proximity of the resource from the places tagged by the user and all his friends.

Chapter 6
Experimental Phase

The experimental phase was divided into two steps. Initially, the performance of the proposed context-aware recommendation engine was tested using some available datasets. Subsequently, according to the proposed system architecture, an application prototype has been realized to promote the cultural tourism of the city of Salerno and the Amalfi Coast. The experimental results obtained through user feedback are satisfactory and show the potential of the proposed system.

6.1 Datasets and evaluation metrics

The ability to analyse the effectiveness of the recommendations given in various contexts is a central issue. Initially, based on the chosen recommendation method and the intended purposes, the behaviour of the developed system will necessarily have to be studied through a set of collected data (datasets): in this case, an analysis will be carried out on the accuracy of the recommendations provided. Later, testing the system in real-time, other properties of the given recommendations will need to be studied in order to improve performance.

6.1.1 Datasets

The possibility to use contextual datasets to analyse the performance of the various methods is crucial.

In this regard, a helpful tool is **CARSKit** (Zheng *et al.*, 2016). This software is freeware and developed in JAVA. It implements state-of-the-art context-aware recommendation algorithms that can be easily configured and tested. It is also convenient to use thanks to an associated guide[1], although a limited number of datasets may be used (Raza and Ding, 2019).

Some of the datasets available for CARS analysis are the following (Ilarri *et al.*, 2018):

- **LDOS-CoMoDa** (Košir *et al.*, 2011): movie dataset including 2296 ratings. The contextual information is divided into twelve contextual dimensions: *time* (morning, afternoon, evening, night), *day type* (working day, weekend, holiday), *season* (spring, summer, autumn, winter), *location* (home, public place, friend's house), *weather* (sunny/clear, rainy, stormy, snowy, cloudy), *social* (alone, my partner, friends, colleagues, parents, public, my family), *emotion* ("endEmo" and "dominantEmo" - sad, happy, scared, surprised, angry, disgusted, neutral), *mood* (positive, neutral, negative), *physical* (healthy, ill), *decision* (user decided which movie to watch, the user was given a movie), *interaction* (first interaction with a movie, n-th interaction with a movie).

- **Frappe** (Shi *et al.*, 2014): dataset collected as part of developing a context-aware mobile application. It contains 96203 ratings of 957 users about 4082 items. The contextual information analysed are six: *daytime*, *weekday*, *homework*, *weather*, *country*, *city*.

- **DePaul Movie** (Adomavicius *et al.*, 2010): dataset consisting of students' ratings in different moments. It contains 97 users, 79 items, and 5043 ratings, of which 1448 ratings without any specified context and 3595 ratings in which the context is specified. The contextual information considered are three: *time* (weekend, weekday), *location* (home, cinema), *companion* (alone, partner, family).

- **InCarMusic** (Baltrunas, Kaminskas, *et al.*, 2011): dataset developed to recommend music to a vehicle's passengers. It contains 4012 ratings of 42 users about 139 items. The contextual information considered are seven: *driving style* (relaxed, sport), *landscape* (coastline, countryside, mountains/hills, urban), *mood* (active, happy, lazy, sad), *natural phenomena* (afternoon, day time, morning, night), *road type* (city, highway, serpentine), *sleepiness* (awake, sleepy), *traffic conditions*

[1] https://arxiv.org/abs/1511.03780

(free road, lots of cars, traffic jam), weather (cloudy, rainy, snowing, sunny);

- **TripAdvisor**: a dataset of recommendations about hotels to users based on a single contextual dimension. This dataset provides information on the *travel type* (family, couples, business, travel only, friends). There are also the user and item features associated with geographical positions. The dataset has the following two versions:

  - **TripAdvisor v1** (Zheng *et al.*, 2012) contains 4669 ratings of about 1202 users and 1890 items;

  - **TripAdvisor v2** (Zheng *et al.*, 2014) contains 14175 ratings obtained by 2731 tourists out of 2269 hotels.

  Finally, there is also a different version containing 28350 ratings about 2371 users and 2269 items.

- **STS** (Braunhofer *et al.*, 2014): dataset containing the data collected through the South Tyrol Suggest app that aims to recommend points of interest. 2534 ratings are available from 325 users on 249 items. The main peculiarity is the number of different contextual dimensions; in fact, the context is obtained by combining fourteen information: *distance*, *time available*, *temperature*, crowdedness, *knowledge of surroundings*, *season*, *budget*, *daytime*, *weather*, *companion*, *mood*, *weekday*, *travel goal*, *means of transport*.

- **TijuanaRestaurant** (Ramirez-Garcia and García-Valdez, 2014): this dataset comprises 1422 ratings of 50 users on 40 items. The users involved had to answer eight questions aimed at both the user's characterization and evaluating the items available. These ratings are provided in both context-free and contextual environments. The contextual dimensions evaluated are two: *time* (weekday, weekend) and *location* (school, home, work).

- **JapanRestaurant** (Oku *et al.*, 2006): dataset in which 938 restaurants are classified. The classification is based on four categories: "is equipped with", "has services of", "recommended for", "environment includes". Fourteen contextual dimensions are divided into four groups:
  - *Time*: month, hour, weekday
  - *Schedule, area type, budget, holiday*

- *Partner*: number of males, number of females, lowest age, highest age, relation, status
- *External facto*r: weather, temperature

6.1.2 Evaluation metrics

Evaluation metrics are crucial tools for the analysis of the goodness of Recommender Systems. These are essential for selecting the appropriate recommendation model in a specific field and comparing the different recommendation techniques and their effectiveness (Shani and Gunawardana, 2011). In fact, several aspects of the recommendation model can be assessed based on the evaluation metric used. In the following, what will be asserted in general for Recommender Systems will also apply to Context-Aware Recommender Systems.

To test the effectiveness of RS or, in the specific case, CARS, the type of experiment to be performed must first be decided.

The simplest and least expensive to implement is the offline experiment that exploits a dataset to simulate users' behaviour interacting with the Recommender System (Gunawardana and Shani, 2009). The behaviour is simulated by storing the actions of the user over time, labelled through the timestamp.

The dataset can be divided into training sets and test sets. For this purpose, one of the most widely used techniques is k-fold cross-validation, which allows randomly partitioning the data set available (Rodríguez *et al.*, 2010). The error will be assessed on each of the k partitions generated by using the remaining parts, in turn, as a training set.

Other types of experiments are user studies and online evaluation (Gunawardana and Shani, 2009). The first consists of recruiting a group of users who must perform some tasks related to interaction with the Recommender System. The second aims to make the RS interact with a more significant number of real users to assess the impact and the effectiveness of the recommendations provided by developing an online testing system (Kohavi *et al.*, 2007).

Once the experiment modalities have been determined, it is necessary to decide the Recommender System' properties to be evaluated.

6.1.2.1 Accuracy Measures

It represents the most used measure in offline experiments and aims to assess the Recommender Systems' accuracy in suggesting items to the user. $T = \{(u, i) : \exists r_{ui}\}$ is the set of user-item pairs on which the test phase is carried out and $\widehat{r_{ui}}$ is the prediction about the pair (u,i). The measurement of the accuracy in the prediction of ratings can be evaluated through (Shani and Gunawardana, 2011):

Mean Absolute Error (MAE): it considers the average of the absolute value of the prediction errors.

$$MAE = \frac{1}{|T|}\sum_{(u,i)\in T}|r_{ui} - \widehat{r_{ui}}| \tag{6.1}$$

Root Mean Square Error (RMSE): it considers the square root of the error square averages and, in this way, larger prediction errors are penalized more than the MAE.

$$\text{RMSE} = \sqrt{\frac{1}{|T|}\sum_{(u,i)\in T}(\widehat{r_{ui}} - r_{ui})^2} \tag{6.2}$$

A different measure to be implemented to assess the accuracy of the Recommender Systems is the one that aims to understand if it is possible to suggest items that the user could use (Gunawardana and Shani, 2009; Shani and Gunawardana, 2011). In this regard, a valid method is the measurement of *precision*, *recall*, and *false-positive rate*. These, knowing the number of elements *true positive* (tp), *false positive* (fp), *true negative* (tn) e *false negative* (fn), shall be calculated as follows:

$$Precision = \frac{tp}{tp + fp} \tag{6.3}$$

$$Recall = \frac{tp}{tp + fn} \tag{6.4}$$

$$FalsePositiveRate = \frac{fp}{fp + tn} \tag{6.5}$$

Other possible measures are the *F-measure* (Van Rijsbergen, 1979; Shani and Gunawardana, 2011):

$$F_\beta = (1+\beta^2) \times \frac{precision \times recall}{(\beta^2 \times precision) + recall} = \frac{(1+\beta^2)tp}{(1+\beta^2)tp + \beta^2 fn + fp} \tag{6.6}$$

and the *Area under the ROC Curve* (Hand and Till, 2001; Shani and Gunawardana, 2011).

Within the Top-N recommendation list, it is possible also to evaluate how many correct items are identified in a list that is provided to the user. Defining *hit* as each item identified precisely, the following assessment methods are presented:

- **Hit Rate (HR)**: let be h_u the number of hits associated with user $u \in U$, the *hit rate* is calculated as follow:
$$\text{HIT RATE} = \frac{1}{|U|} \sum_{u \in U} h_u \tag{6.7}$$
- **Average Reciprocal Hit Rate (ARHR)**: let be n the total number of hits and r_i $i = 1, \dots, n$ the rank of i-th hit, the average reciprocal hit rate is calculated as follow:
$$ARHR = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{r_i} \tag{6.8}$$
  In this way, the hits far from the first positions of the Top-N list are penalized.
- **Cumulative Hit Rate**: the hits whose forecast is below a set threshold are excluded. In this way, items that the user would not appreciate are not considered by the evaluation.

Normalized Distance Based Performance (NDBP) (Yao, 1995), and the **R-score metric** (Shani and Gunawardana, 2011) are also functional. In particular, the latter makes it possible to assess the usefulness of the recommendations made.

6.1.2.2 Other Measurements

There are many other possible analysable properties of a Recommendation System.

The *coverage* aims to measure the percentage of recommendable items. A good value may indicate the system's ability to suggest new items to appropriate users quickly.

It is also possible to measure: The reliability of the recommendations provided (*confidence*), the *trust* the user has in the RS, the ability to balance between the recommendations of known items and items not known to the user (*novelty*), the *serendipity* (C. D. Wang *et al.*, 2019), the *diversity* of recommended items and the stability of the recommendations provided in the presence of false information (*robustness*).

Finally, we can also evaluate:

- *Adaptivity*: the Recommender Systems' ability to adapt to changes in the user behaviour or item value over time.

- *Scalability*: the Recommender Systems' ability to process more data with reasonable memory occupation and slowdown.

6.2 First Step: The Context-Aware Recommendation Engine

In this section, the numerical results of the proposed approach on different datasets will be presented. In particular, the accuracy of the De Paul Movie (Zheng *et al.*, 2016), TripAdvisor v1 (Zheng *et al.*, 2012), and TripAdvisor v2 (Zheng *et al.*, 2014) datasets will be evaluated. Finally, the LDOS-CoMoDa dataset will be analysed to perform an evaluation based on a more significant number of contextual information.

In particular, the Matrix Factorization technique used is based on the Singular Value Decomposition and the Biased Stochastic Gradient Descent methods (Gene H. Golub, 2013) with the following values for the required parameters obtained experimentally:

- $Latent\ factors\ =\ 5$
- $\alpha =\ 0.0001$
- $\lambda\ = 0.0005$
- $Iterations\ =\ 20$

6.2.1 Accuracy Evaluation

As previously mentioned, to initially evaluate the proposed recommendation engine, an "offline" experiment (Shani and Gunawardana, 2011) is presented on some available datasets: the ability of the proposed approach to make rating forecasts on De Paul Movie and TripAdvisor datasets (Table 6.1) is evaluated.

Table 6.1 *Contextual information of chosen datasets*

DePaul Movie		
Location	*Companions*	*Time*
Home	Alone	Weekend
Cinema	Family	Weekday
	Friends	

TripAdvisor
Travel type
Family
Couples
Business
Travel only
Friends

The Mean Absolute Error and the Root Mean Squared Error will be considered as estimates for prediction errors to assess accuracy. For this purpose, it is indicated with $T = \{(u,i): \exists r_{ui}\}$ the set of elements on which the errors are calculated, while with $\widehat{r_{ui}}$ is indicated the rating forecast made by the method used.

Table 6.2 shows the results obtained through the proposed approach on the De Paul Movie dataset as the nodes taken into consideration vary. Table 6.3 instead shows the results on the TripAdvisor dataset. In particular, the technique of k-fold Cross-Validation (Rodríguez *et al.*, 2010) with $k = 5$ was used.

Table 6.2 *Numerical Results of the proposed approach on De Paul Movie dataset*

DePaul Movie dataset		
Nodes	**MAE**	**RMSE**
Time	0.9419	1.1823
Location	0.9073	1.1425
Companions	0.8410	1.0909
Time-Location	0.8153	1.0671
Time-Companions	0.7463	0.9988
Location-Companions	0.7257	0.9802
Time-Location-Companions	0.6591	0.9186

Regarding the De Paul Movie dataset, it is interesting to notice that the results obtained by evaluating only the size node *Companions* return values very similar to the results obtained on the pair of size nodes *Time-Location*. It can be noted, above all, that the first case presents three possible contextual parameters (Alone, Family, Friends). In contrast, the second case presents four (Weekend-Home, Weekend-Cinema, Weekday-Home, Weekday-Cinema). It can be observed that the system's performance, when all the values that the dimension nodes can assume are known, seems to depend not so much on the number of contextual dimensions considered but on the number of values that the entire context (Embedded Context) can assume. This aspect will be evaluated in depth through the use of the LDOS-CoMoDa dataset.

Table 6.3 *Numerical Results of the proposed approach on TripAdvisor datasets*

TripAdvisor v1 dataset		
Node	**MAE**	**RMSE**
Travel type	0.9950	1.3043

TripAdvisor v2 dataset		
Node	**MAE**	**RMSE**
Travel type	0.9314	1.2172

The results obtained show the quality of the proposed approach (Table 6.4): the introduction of the user and item bias associated with the entire context, rather than to the single values that each contextual dimension can assume, returns exciting results in rating prediction (Figure 6.1). Indeed, the errors obtained from the proposed approach are better than ones of comparison methods.

Table 6.4 *Comparison between the proposed approach and other CARS techniques on De Paul Movie dataset*

DePaul Movie dataset		
Approach	**MAE**	**RMSE**
CAMF-CI	0,7122	0,9660
CAMF-CU	0,6682	0,9198
CAMF-C	0,7029	0,9571
CAMF-CUCI	0,6753	0,9206
ItemSplitting-BiasedMF	0,7185	1,0362
UserSplitting-BiasedMF	0,7361	1,0588
The proposed approach	0.6591	0.9186

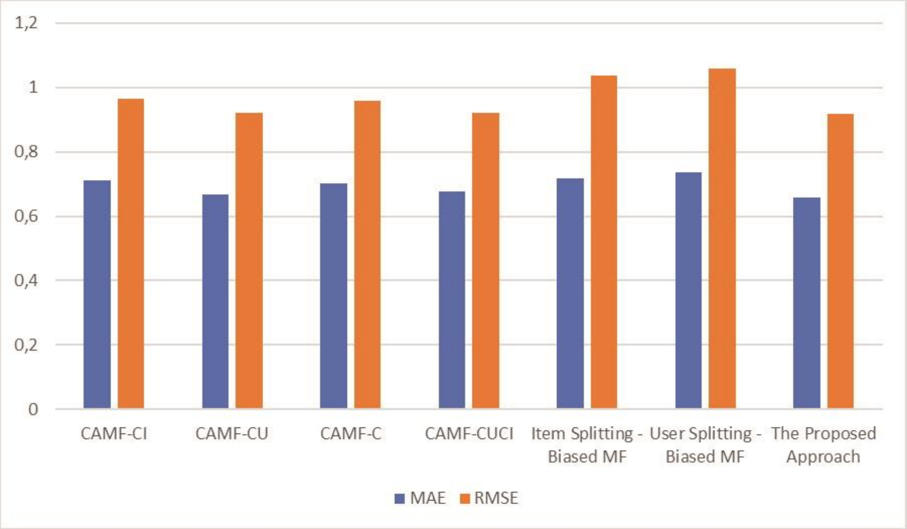

Figure 6.1 *Graphical representation of the results of the methods on the De PaulMovie dataset*

6.2.2 The Sparsity Problem

Previously the concept of Embedded Context was presented. The purpose was to emphasize using the context as it can be represented in the Embedding Space. In other words, rating forecasts can be obtained by avoiding to consider the single contextual dimensions separately. In fact, the sum of the possible values that these dimensions may assume does not necessarily lead back to the precise context in which a user is located.

The limit of such analysis is, however, evident: as the contextual information increases, the number of possible values, that the Embedded Context can assume, are obtained through the product of the number of possible values of each contextual dimension:

$$\left.\begin{array}{l} Contextual\ information\ c_1\ has\ l_{c_1} \\ \vdots \\ Contextual\ information\ c_t\ has\ l_{c_t} \end{array}\right. \Rightarrow \prod_{i=1}^{t} l_{c_i}\ Embedded\ Contexts\ v$$

For this purpose, it is necessary to make a selection of the contextual information to analyze.

In this regard, an analysis of the LDOS-CoMoDa dataset (Kosir, 2012) is instrumental. It is composed of 2296 ratings of 121 users on 1232 movies divided as follows:

- 44 ratings in which the context is not specified;
- 2252 ratings in which contextual information is identified.

This dataset comprises 12 types of contextual information (Figure 6.2):

1. $Time$ (4): Morning, Afternoon, Evening, Night
2. $Day\ Type$ (4): Working day, Weekend, Holiday
3. $Season$ (4): Spring, Summer, Autumn, Winter
4. $Location$ (3): Home, Public place, Friend's house
5. $Weather$ (5): Sunny, Rainy, Stormy, Snowy, Cloudy
6. $Social$ (7): Alone, My partner, Friends, Colleagues, Parents, Public, My family
7. $Emotion$ (7): Sad, Happy, Scared, Surprised, Angry, Disgusted, Neutral
8. $Dominant\ Emotion$ (7): Sad, Happy, Scared, Surprised, Angry, Disgusted, Neutral
9. $Mood$ (3): Positive, Neutral, Negative
10. $Physical$ (2): Healthy, Ill

11. $Decision$ (2): User decided which movie to watch, a user was given a movie

12. $Interaction$ (2): First interaction, n-th interaction

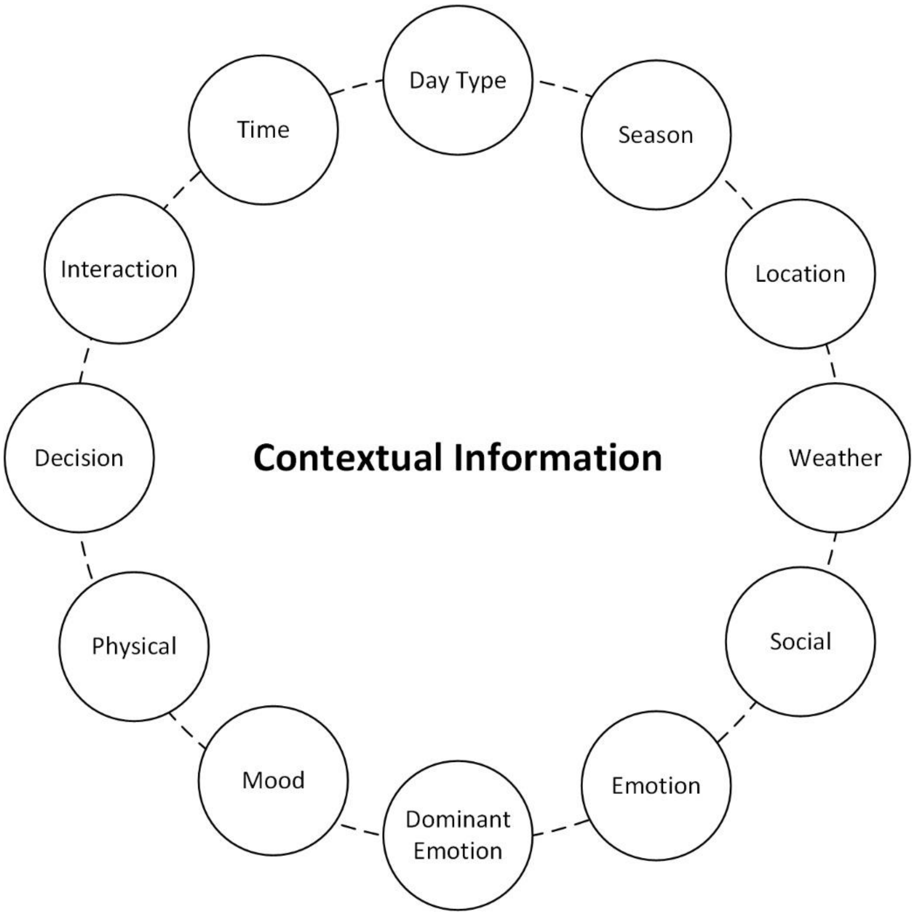

Figure 6.2 *Contextual information in the LDOS-CoMoDa dataset*

First of all, through the LDOSCoMoDa dataset, the extent to which the Embedded Context suffers from the sparsity problem on bias was studied because the values assumed by context are not known for all ratings. The results of this analysis are shown in Figure 6.3. To address this problem, the lack of information has been introduced as a possible concept node of the related dimension node, thus increasing the number of possible embedded contexts: this poses a challenge to the proposed approach but also allows to evaluate its behaviour in an unfavourable case.

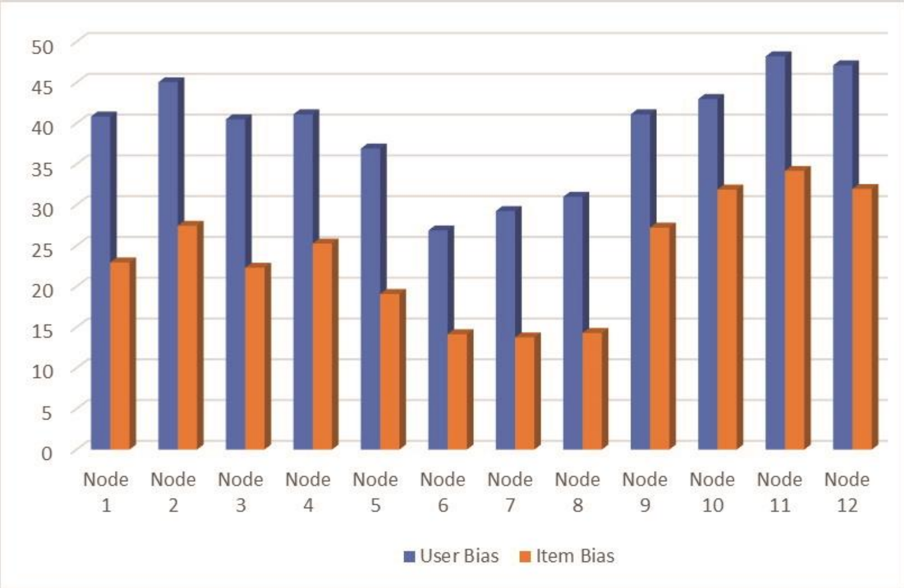

Figure 6.3 *Mean of the percentage of known user bias and known item bias in LDOSCoMoDa dataset during a training phase*

Figure 6.3 shows that the dimension nodes with the lowest number of concept nodes have, on average higher percentages of filling the bias matrices for both users and items during the training phase of the k-fold Cross-Validation. Moreover, the percentages averages of known bias decrease significaly when more dimension nodes are taken at the same time.

In this regard, a phase has been implemented in the proposed method where averages replace the unknown biases for concept nodes:

$$\forall i \in I \; if \; (b_{ic} = 0) \; then \; b_{ic}^{new} = \frac{1}{|\{b_{lc}: b_{lc} \neq 0\}|} \sum_{s=1}^{tot-item} b_{sc} \qquad (6.9)$$

$$\forall u \in U \; if \; (b_{uc} = 0) \; then \; b_{uc}^{new} = \frac{1}{|\{b_{zc}: b_{zc} \neq 0\}|} \sum_{s=1}^{tot-user} b_{sc} \qquad (6.10)$$

This method improvement was also used for the numerical results previously presented.

Moreover, due to the sparsity problem, the proposed approach was tested using different contextual information divided according to the number of values each dimension node can assume (number of concept nodes). In particular, the contextual information was divided as follows:

- Nodes 9-Mood(3), 10-Physical(2), 11-Decision(2) and 12-Interaction(2): nodes with a low number of values available

- Nodes 1-Time(4), 2-Daytype(4), 3-Season(4), 4-Location(3): nodes with an intermediate number of values available

- Nodes 5-Weather(5), 6-Social(7), 7-Emotion(7), 8-DominantEmotion(7): nodes with a high number of values available

The results obtained according to the presented partition are shown in Table 6.5.

Table 6.5 *Numerical results of the experiments tested on LDOS-CoMoDa dataset*

LDOS-CoMoDa dataset				
Nodes	**#DimensionNodes**	**#ConceptNodes**	**MAE**	**RMSE**
10	1	2	0.8344	1.0877
10-11	2	4	0.8643	1.0967
10-11-12	3	6	0.8288	1.0764
9-10-11-12	4	9	0.8281	1.0826
1	1	4	0.8493	1.0990
1-2	2	8	0.8621	1.1159
1-2-3	3	12	0.8856	1.1457
1-2-3-4	4	15	0.8907	1.1440
8	1	7	0.7380	0.9711
7-8	2	14	0.7265	0.9556
6-7-8	3	21	0.7352	0.9646
5-6-7-8	4	26	0.7695	0.9992
All	12	50	0.8386	1.0680

Table 6.5 shows that increasing the number of concept nodes does not always improve rating prediction. This aspect is further confirmed by the result obtained by considering all nodes reporting MAE and RMSE values that are worse than or similar to the respective values obtained considering individual dimension nodes.

This aspect allows us to state that it is unnecessary to use a large number of dimension nodes and, therefore, contextual information to obtain reliable results through the proposed method in the presence of sparsity. In other words, the limitation due to the exponential increase of the Embedded Context is compensated by the fact that the method does not improve in the case of sparsity by considering a larger number of dimension nodes.

In Table 6.6, there are instead the results of all available dimension nodes taken individually.

Table 6.6 *Results on individual nodes of the LDOSCoMoDa dataset*

Contextual Information	#Contexts	MAE	RMSE
Time	4	0.8493	1.0990
DayType	4	0.8568	1.1012
Season	4	0.8307	1.0738
Location	3	0.8490	1.0923
Weather	5	0.8569	1.1130
Social	7	0.8608	1.1107
Emotion	7	0.7380	0.9711
Dominant Emotion	7	0.7374	0.9771
Mood	3	0.8515	1.1008
Physical	2	0.8344	1.0877
Decision	2	0.8408	1.0799
Interaction	2	0.8253	1.0630

Table 6.6 shows that, in the case of sparsity, the node dimensions with a more significant number of children (concept nodes) do not always lead to better accuracy of the Context-Aware Recommender System. In fact, we can see that the Location nodes (3 concept nodes), Physical and Interaction (2 concept nodes) present better MAE and RMSE than the Social node (7 concept nodes).

We also note that the nodes related to Emotion and Dominant Emotion present the best accuracy results, as can be seen in (Zheng *et al.*, 2013).

This analysis allows considering the introduction of *tolerances* on errors values given by single nodes. For example, in LDOS-CoMoDa, the tolerance values $MAE_{tol} = 0.75$ and $RMSE_{tol} = 1.00$ are have been set. The nodes that give errors below these tolerances are taken. The nodes Emotions and Dominant Emotions are the only ones that have this property.

This solution can be implemented in a real scenario that presents sparsity in order to select only the right context dimensions. If the error analysis is done periodically, the system performance will be increased over time through newly acquired information. Moreover, context management tools, such as CDT, significantly reduce sparsity's impact on the system by selecting only doable contexts. Nevertheless, an in-depth study of the effects of the sparsity problem had to be done in order to increase the awareness of the potential and limits of the proposed approach.

Finally, in Table 6.7, the comparison with some known methods through the returned RMSE value is presented. The proposed approach was evaluated through the nodes Emotions and Dominant Emotions following the previous study that allowed the introduction of tolerance. The comparison results are acquired from (Zheng *et al.*, 2013).

Table 6.7 *Numerical results of different approaches with LDOSCoMoDa dataset*

Method	Considered Nodes	RMSE
CAMF-C	Emotion and Dominant Emotion	1.012
CAMF-CI	Emotion and Dominant Emotion	1.032
CAMF-CU	Emotion and Dominant Emotion	0.932
Proposed Approach	Emotion and Dominant Emotion	0.891

6.3 Second Step: An Application Prototype

Based on Figure 5.1, a simple application prototype was developed: a hybrid mobile application, designed and implemented, together with a server-side component. In particular, the client was developed through the Ionic and Apache Cordova frameworks, using the main Web languages (HTML5, CSS and Javascript). Concerning the proposed architecture, it mainly represents the Application Layer. The server-side component, on the other hand, was implemented mainly in Python. It refers to the software needed to implement the recommendation engine and the primary API services used by the mobile application. The APIs that the environment provides make the wealth of information available to any service compatible with a REST and JSON-based approach. Finally, the Laravel framework was used to create the Management Module, which enabled data management. In this regard, the recently adopted No-SQL approach and the latest technologies will allow new data structures to be easily integrated into the Knowledge Base.

The App was designed to support tourists visiting Salerno, a city in the Campania region of Italy, and the Amalfi Coast. In this phase, the primary services and API contents potentially useful for tourists have been identified. The App also collects information from social environments by adapting the proposed itinerary, taking into account the user's communities and interests.

1. *Services definition*

   The definition phase of the services was implemented following three main operations: obtaining information on the current location, searching for resources in the area, and acquiring the user's profile and interests. To this end, some Web services accessible through APIs were identified and enriched, where possible, by specific services implemented for the city of Salerno and the Amalfi Coast. These services mainly exploit the data from the Content Providers inserted in the Knowledge Base and updated through a Web management platform (Management Module).

In the beginning, the user's location is determined by the GPS sensor on the mobile device. In particular, it is necessary to use a Reverse Geocoding service, i.e. the process to turn geographical coordinates into a "human-readable" address, made available by the Google Maps API. Then, to obtain the location's description, based on the position detected, it was decided to use the information from website pages, specifically the RESTful Web services API: convenient access to services, data and metadata through HTTP requests. These pages are often associated with coordinates of reference and the proposed approach exploits this property: when searching for a page related to a country or a city, the one closest to the user's coordinates is obtained. In particular, the description of the Province of Salerno can be enriched by integrating some ad hoc services that recall the information shared by Content Providers and suitable for specific types of users.

The story to be presented is then enhanced by inserting other users' experiences, using mainly the comments released on TripAdvisor. In this regard, resources are searched using the "map" method of the TripAdvisor API, which allows you to search for attractions (museums, sports facilities, shops, ...), hotels and restaurants through, respectively, the methods "map_attractions", "map_hotels", "map_restaurant" and, more generally, the method "location_mapper". Concerning the Province of Salerno, the user can also obtain the main points of interest in the Knowledge Base and managed by Content Providers using the Management Module.

Finally, to overcome the initial cold start problem, it is necessary to collect data relating to users. The main objective is to access the user's social data to retrieve, for example, his profile, his LIKEs, his friends, his favourite events and places: this is done through the use of the Facebook API and the analysis of some main methods, such as "user_likes", "user_friends", "user_events" and "tagged_places". In particular, the "user_likes" permission provides a list of all the Facebook pages a user has liked, while the "user_friends" permission provides a list of the person's friends who use the app. Concerning users' likes, the category attribute associated with social network pages was mainly used. These data are then integrated and updated with implicit data, obtained from the user's behaviour, and explicit data, obtained through the manual updating of the user's profile (Knowledge-Based Approach).

2. Recommendation model: a hybrid approach

The hybrid recommendation model is following what was presented in the previous chapter. In particular, at this stage, the use of CDT allows both to represent all possible contexts and to consider only the doable contexts (as defined in Chapter 1). For instance, it avoids context elements that do not change the global view of the context itself: the use of CDT allows the contextual recommendation engine to avoid necessarily perceiving the whole context, but to react only to the significant contextual dimensions. In this way, many of the problems discussed earlier, such as sparsity, can be avoided, and it is possible to obtain more accurate real-time ("online") predictions than "offline" ones.

3. *Presentation of information*

The last phase is based on the presentation of contextualised and personalised information: the result is presented in the form of a well-organised story. This story is able to provide a general introduction to the place reached by the user, enriched by the experiences shared by other users, a list of the main attractions, suggested information on nearby places visited by friends and ancillary services. In particular, the main points of interest can be reached by following the personalised paths.

When a new user starts the App, if he does not have a Facebook account or has not yet logged in, he gets information about the place and the main points of interest that he can view on the map and reach by a navigator. Instead, once logged in the system inserts some recommended points of interest based on the user's preferences, obtained through an analysis of Facebook data. Finally, if some of the user's friends have been in the vicinity of the current location, the system shows what they have visited.

Some screenshots of the application, as described above, are shown in Figure 6.4. The user automatically gets a tale of the place where he is located, including a list of the main points of interest recommended based on his profile, some places in the surroundings visited by his closest friends and the primary services designed for his needs. The user can delve into a specific place's history and get personalised visit paths using the data and services managed through the Management Module. Figure 6.5 shows a screenshot of the platform that allows Content Providers to power and manage the Knowledge Base and build new services to provide the user.

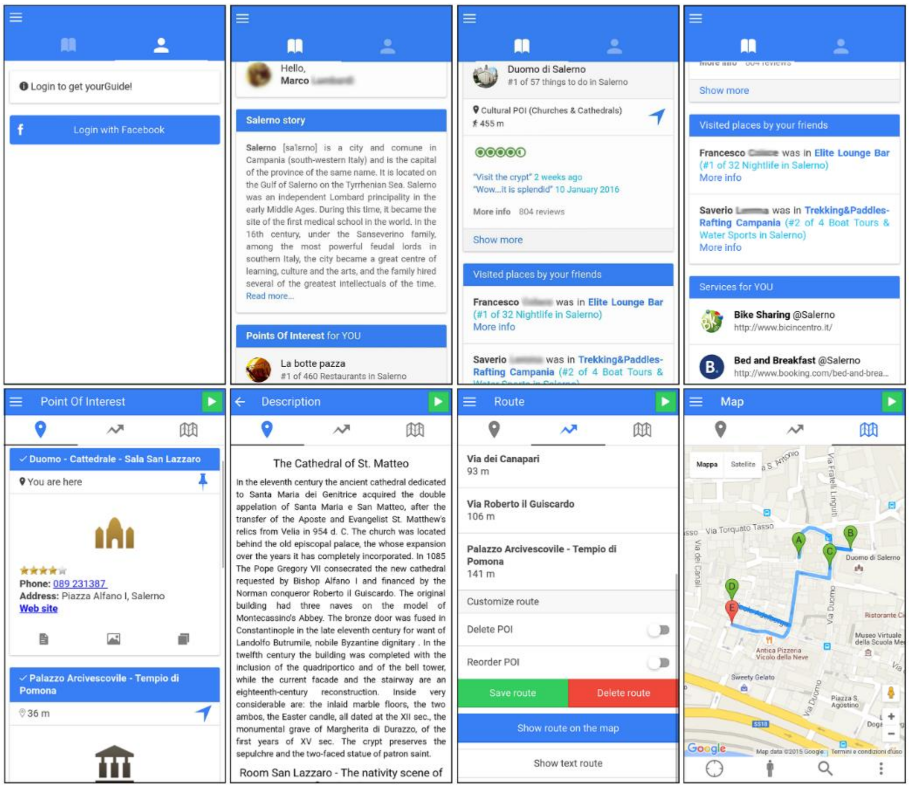

Figure 6.4 *Mobile application screenshots*

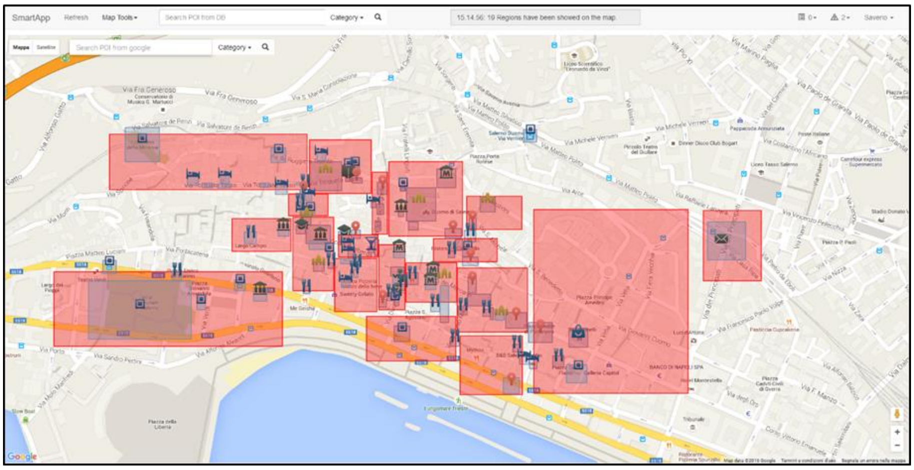

Figure 6.5 *Management Module screenshots*

6.3.1 Experimental Results

One of the main experimental campaigns conducted involved 2300 participants: tourists visiting Salerno and Amalfi Coast between 18 and 58, who do not know the study subject. Everyone is registered on the social network Facebook and owns a mobile device. The App prototype has been distributed and installed by all participants who have logged in with their Facebook credentials. After interacting with the application, participants responded, according to the Likert scale, to five statements, summarized below: (A) The system provides correct information on the place of a visit, based on personal preferences and current context; (B) The system managed to adapt to changes in context; (C) The services offered are adequate and meet the needs of the tourist; (D) The information has been adequately presented, in the form of a modern tourist guide; (E) The integration of experience of friends and other users has been useful to improve the App performance. To each statement, five possible answers were associated: "Totally disagree" - TD, "Disagree" - D, "Undecided" - U, "Agree" - A, "Totally Agree"- TA.

Table 6.8 shows the results of the questionnaire carried out by the participants at the end of using the implemented prototype. As it can see, observing all the answers, the degree of user satisfaction is high. In particular, aggregating the positive responses (Agree and Totally Agree), a percentage higher than 80% is reached for each statement.

Table 6.8 *Experimental results*

Statement	**Answer**				
	TD	**D**	**U**	**A**	**TA**
A	100	65	91	959	1085
B	119	32	75	1118	956
C	172	56	99	952	1021
D	141	68	105	879	1107
E	195	67	104	977	957

As shown in Figure 6.6, the degree of user satisfaction is evidenced by the separation between the histogram bars related to positive answers and those relating to negative or neutral answers. As can be seen, particularly for answer E, even if the percentage is not significant, there is a group of users who totally disagree with the prototype usage experience. This result seems to suggest that it might have been difficult, for a reduced percentage of users, to enter into the perspective of using the implemented prototype. However, this feedback poses additional challenges to improve the entire system.

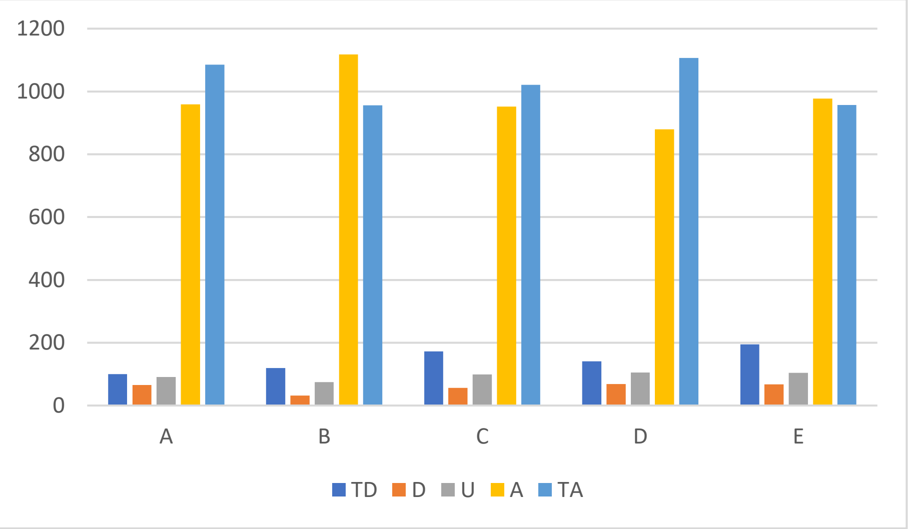

Figure 6.6 *Analysis of questionnaires*

Finally, in Table 6.9, the analysis of the answers to the statements shows great results. Overall user satisfaction can be seen through the percentage of positive responses, which reaches around 87% on average. Our prototype peaked with satisfaction (about 90%) about the system's activity at context changes (statement B). This result suggests that users have well perceived the degree of Context Awareness of the system.

Table 6.9 *Analysis of results*

Statement	Percentage		
	Negative	Neutral	Positive
A	7,17%	3,95%	88,88%
B	6,57%	3,26%	90,17%
C	9,91%	4,31%	85,78%
D	9,09%	4,56%	86,35%
E	11,39%	4,52%	84,09%

Chapter 7
Future Works

In Chapter 5, the system architecture was presented and the research activity mainly concerned the Recommendation Engine based on Context Awareness. In this regard, future work may focus on studying and developing the other modules of the architecture.

In particular, it started by introducing a chatbot, called CHAT-Bot (Cultural Heritage Aware Teller-Bot), based on the previously proposed Context-Aware System. According to the profiles and behaviours of tourists and their context, this chatbot can recommend contents and services through "friendly" approaches: use of natural language in human-computer interaction is attractive because it is one of the most flexible, efficient, and natural means to communicate (Casillo *et al.*, 2020).

The process that allows the machine to understand and relate to human beings is Natural Language Processing (NLP), which Machine Learning often joins to give it a learning capacity. In particular, NLP has extended its study range beyond the single word to include entire sentences. This approach makes it possible to deal more effectively with the disambiguation problems that often arise. In fact, it is not uncommon to come across words that may have more meanings. The mapping of words and sentences has made it possible to avoid these pitfalls, allowing, for example, to assign every word the sense most consistent with the context in which it is inserted.

In recent years, there has been a shift from the purely conversational world to that of chatbots as a multimedia interface, in which text, images and command buttons coexist (Clarizia *et al.*, 2018). In particular, there are many recent literature studies on the implementation of chatbots as support for humans in various application fields. Such systems are designed using Natural Language Processing techniques, such as *sentence-classification*, *key-concept identification* and *recurrent neural network*. (Haller and Rebedea, 2013; Sun and Zhang, 2018; Cerezo *et al.*, 2019; Zalake and Naik, 2019).

The proposed architecture about the chatbot is based on some main modules, as shown in Figure 7.1, which will be integrated with the architecture modules proposed in Figure 5.1: Context-Aware Module, Storytelling Module, etc. (in practice, following the same approach described above).

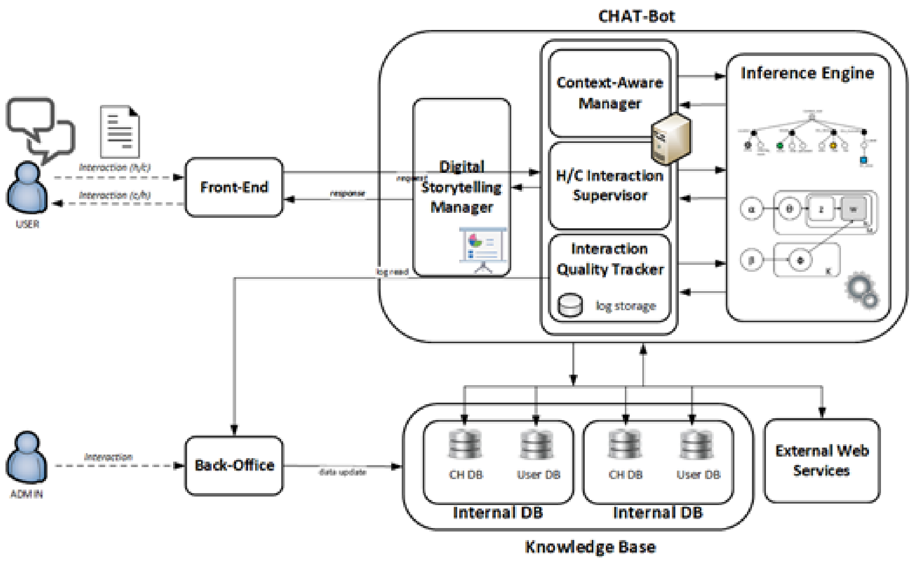

Figure 7.1 *Proposed architecture for a chatbot to be integrated into the designed system for supporting touristic experiences*

The **Storytelling Module** is closely related to the bot ability to guide the user through the whole experience, making the way of proceeding while leaving the user free to express himself and immerse himself in the personalised story of a place.

Each user acts differently from the others, becoming part of a creative process and creating a unique and unrepeatable visit. In planning an itinerary and designing the narration, therefore, not all progress must be defined, but a plurality of scenarios must be prepared that the user can explore freely up to crucial moments, common to all, or almost all, of the scenarios.

The **Context-Aware Manager** deals with representing all the possible Contexts of use through the Context Dimension Tree and performing contextualised queries. In this way, it is possible to extract and provide personalised information by aggregating and custom-tailoring data and services extracted from different sources. Some resources are private (internal resources) or managed directly by the chatbot provider. These resources can be used, for example, to maintain the profiles of registered users or the data of the museums reviewed and the promotions offered to users. The chatbot can also interface with external services (external resources), for example, for booking an art exhibition.

Other modules, such as the **Human/Computer Interaction Supervisor** and the **Interaction Quality Tracker**, have the following objectives:

- supervise the dialogue, keeping track of the timing of interaction, identifying ambiguous questions or dialogue sessions that are not convergent or too long, and so on;

- carry out monitoring interactions between the user and the chatbot, producing synthetic quality indicators and highlighting critical aspects helpful in improving the system.

The core of the architecture is the **Inference Engine**, which includes *text analysis* and *context extraction*. It is assumed that the text generated by a chat is a mixture of contextual information. Some words help define the different Context Elements useful in searching for the same Context that can be identified through the Context Dimension Tree (CDT).

Latent Dirichlet Allocation (LDA) is a model suitable for this purpose. It can be used to explain the correlation between keywords and topics (in this case, Context Elements), as shown in the following Figure 7.2.

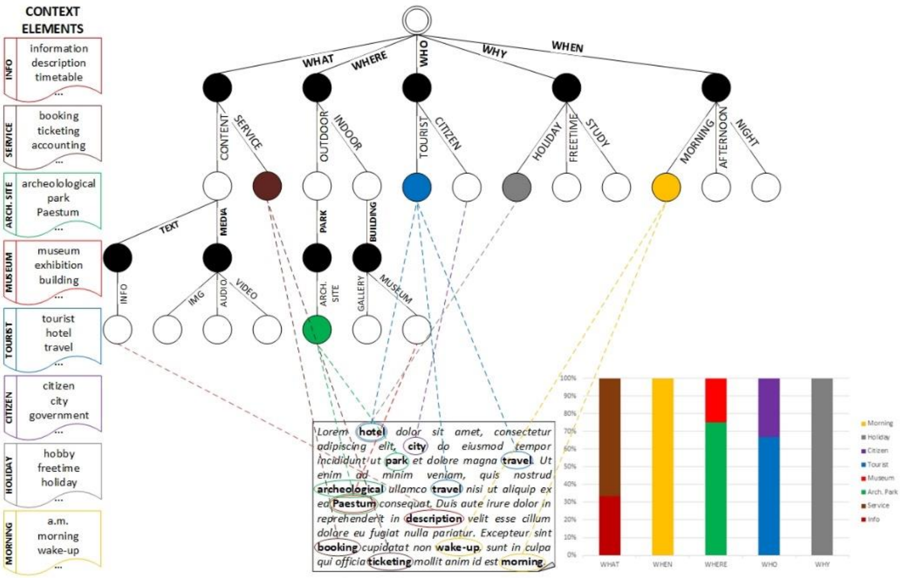

Figure 7.2 *Proposed approach for the definition of Context Elements*

Through textual analysis, it is possible to know the user (citizen or tourist), where he is or where he would like to be (museum, archaeological park, etc.), the purpose of his visit (holiday, leisure, or study) and what they need: services to book or buy tickets, information about timetables, multimedia contents, etc.

For example, a possible Context could be a tourist who wants to get services to access an archaeological park during the holiday. First, through the analysis of the chat and then through the elaboration of the current Context, the chatbot can define the real intention of the user in order to satisfy his needs better or to recommend specific services properly.

In practice, the user's interaction with the chatbot is divided into shorter and simpler sentences (clusters) through appropriate Bayesian filters for keywords, assuming that each sentence is semantically related to the other. Therefore, the proposed approach is based on the following assumption: the probability that the word W belongs to the concept node N_c, within the CDT, is proportional to how much a topic (e.g., the purpose of a tourist visit to an art city) has already been treated and the number of times that a word has been used about the specific topic. This model automatically characterises chats without the need to specify the semantic value of the text's words.

Furthermore, using the LDA approach on a set of chats that belong to the same domain (in the case analyzed, tourism), it is possible to extract a Mixed Graph of Terms (mGT) automatically. It can be used both for the design of the tree of Context and the constraints associated with it is to detect the Context extracted in real-time from the user's chat with the bot (Colace, De Santo, Greco, Amato, *et al.*, 2014).

In particular, LDA was mainly used to generate topics within chats (text documents). The system processed these topics as Context Elements during the Context Dimension Tree use.

7.1 Research and use of the Context through mGT and CDT

A complex structure like the mGT can capture and represent the contextual information in a set of chats that belong to a specific domain (for example, tourism). This graph can be extracted automatically and used for the classification of the text or to label the concept nodes N_c Furthermore, it is useful to know which of the nodes participate in the *context definition* at a given time. Formally, mGT can be defined as a graph $g = < N, E >$ where:

- N = {R, W} is a finite set of nodes, whose elements can be aggregates or aggregators
- $E = E_{RR}, E_{RW}$ is a set of arcs that connect the aggregates and aggregators

The proposed approach is essentially composed of two primary modules located within the Inference Engine: a module for constructing the Mixed Graph of Terms and a module for extracting the Context Elements, as shown in Figure 7.3 (a).

- **Mixed Graph of Terms building module**: this module builds the mGT starting from a set of documents that belong to a specific domain (tourism) and have been previously labelled according to the contextual information. The mGT can also be used in the design phase of the Context Dimension Tree.

- **Context Mining Module**: this module involves extracting the Context or the different Context Elements, thanks to mGT as a *context filter*. This module's input consists of a generic chat, the mGT extracted and the CDT about the specific domain. The output is the Context related to the chat.

Each Context Element is associated with a dedicated section of the database containing relevant and specific data. The contextual query is performed automatically by defining a global view given by partial views. In addition to simple data, the same mechanism can be used to select useful services related to the identified Context, as shown in Figure 7.3 (b).

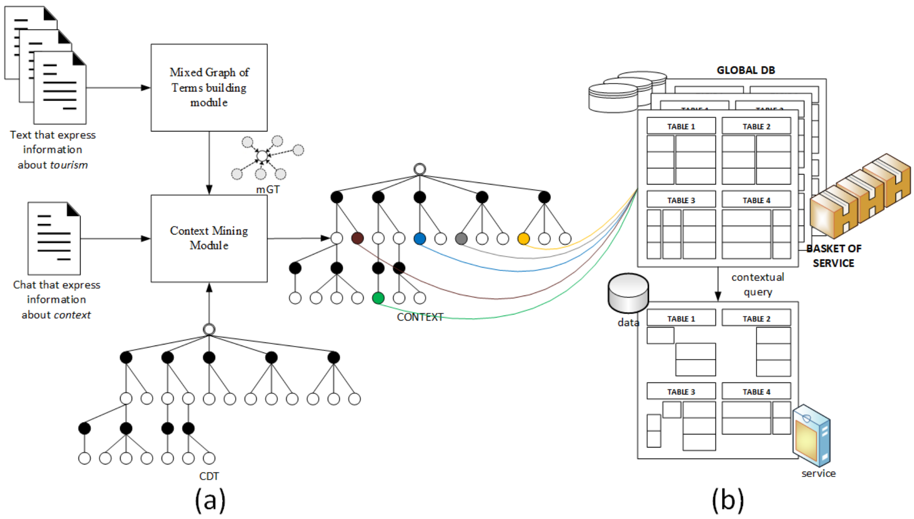

Figure 7.3 *Use of the Context defined through mGT and CDT: an example*

Based on the proposed approach, an early prototype was developed through the telegram API (Figure 7.4). In this way, the proposed methodology will be tested separately and improved before being integrated into the overall system proposed in Figure 5.1.

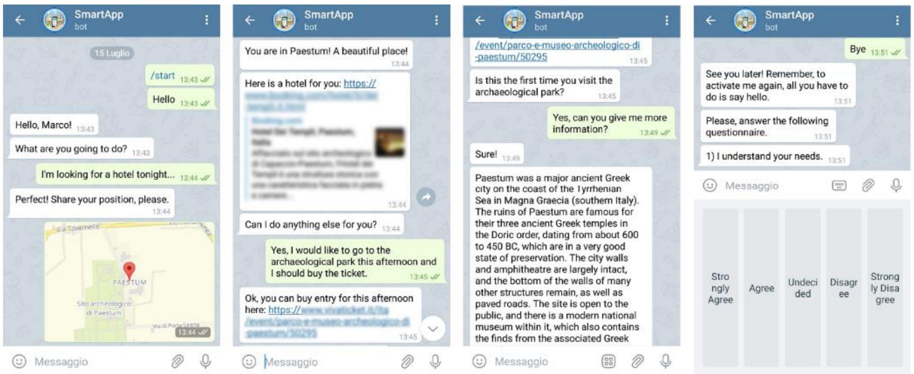

Figure 7.4 *Some screenshots of the chatbot prototype*

Conclusions

Real-time knowledge of the Context where the user is located allows recommending highly personalised services and considers numerous aspects. This research work aimed to deepen the logic of integration between Recommendation Techniques and Context Awareness, and the interoperability between heterogeneous platforms (existing or new), in order to allow the automatic and adaptive to the Context construction of highly tailored as well as complete services, capable of going beyond the information phase, by facilitating tourists in every moment of their experience: from the search for a destination to the definition of the contents, up to the purchase, use and commentary of their experience, integrating services of tourism promotion, booking, e-ticketing, e-commerce, social networking, etc.

In this scenario, the proposed architecture can be declined in different mobile applications able to behave like a modern tourist guide and follow the user at every stage, dynamically recommending personalised and contextualised paths based on a large number of variables or unforeseen events that may occur during travel. The experimental results show the ability of the system to be effective. Future activities include improving the developed prototype, including a chatbot, and an experimental campaign involving a more significant number of users.

References

Aberger, C. R., Koren, Y., Bell, R. and Volinsky, C. (2009) 'Recommender : An Analysis of Collaborative Filtering Techniques', *Computer*.

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M. and Steggles, P. (1999) 'Towards a better understanding of context and context-awareness', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/3-540-48157-5_29.

Adomavicius, G. and Tuzhilin, A. (2015) 'Context-aware recommender systems', in *Recommender Systems Handbook, Second Edition*. doi: 10.1007/978-1-4899-7637-6_6.

Adomavicius, G., Tuzhilin, A., Berkovsky, S., De Luca, E. W. and Said, A. (2010) 'Context-awareness in recommender systems: Research workshop and movie recommendation challenge', in *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems*. doi: 10.1145/1864708.1864801.

Albanese, M., D'Acierno, A., Moscato, V., Persia, F. and Picariello, A. (2011) 'A multimedia semantic recommender system for cultural heritage applications', in *Proceedings - 5th IEEE International Conference on Semantic Computing, ICSC 2011*. doi: 10.1109/ICSC.2011.47.

Alexander, B. and Levine, A. (2008) 'Web 2.0 Storytelling: Emergence of a New Genre', *Educause Review*.

Ashton, K. (2009) 'That Internet of Things Thing', *RFID Journal*.

Atzori, L., Iera, A. and Morabito, G. (2010) 'The Internet of Things: A survey', *Computer Networks*. doi: 10.1016/j.comnet.2010.05.010.

Badii, A., Crouch, M. and Lallah, C. (2010) 'A Context-Awareness Framework for intelligent networked embedded systems', in *Proceedings - 3rd International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services, CENTRIC 2010*.

doi: 10.1109/CENTRIC.2010.29.

Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., Lüke, K. H. and Schwaiger, R. (2011) 'InCarMusic: Context-aware music recommendations in a car', in *Lecture Notes in Business Information Processing*. doi: 10.1007/978-3-642-23014-1_8.

Baltrunas, L., Ludwig, B. and Ricci, F. (2011) 'Matrix factorization techniques for context aware recommendation', in *RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems*. doi: 10.1145/2043932.2043988.

Baltrunas, L. and Ricci, F. (2009) 'Context-based splitting of item ratings in collaborative filtering', in *RecSys'09 - Proceedings of the 3rd ACM Conference on Recommender Systems*. doi: 10.1145/1639714.1639759.

Baltrunas, L. and Ricci, F. (2014) 'Experimental evaluation of context-dependent collaborative filtering using item splitting', *User Modeling and User-Adapted Interaction*. doi: 10.1007/s11257-012-9137-9.

Barthes, R. and Duisit, L. (1975) 'An Introduction to the Structural Analysis of Narrative', *New Literary History*. doi: 10.2307/468419.

Bartolini, I., Moscato, V., Pensa, R. G., Penta, A., Picariello, A., Sansone, C. and Sapino, M. L. (2016) 'Recommending multimedia visiting paths in cultural heritage applications', *Multimedia Tools and Applications*. doi: 10.1007/s11042-014-2062-7.

Batet, M., Moreno, A., Sánchez, D., Isern, D. and Valls, A. (2012) 'Turist@: Agent-based personalised recommendation of tourist activities', *Expert Systems with Applications*. doi: 10.1016/j.eswa.2012.01.086.

Bobadilla, J., Ortega, F., Hernando, A. and Gutiérrez, A. (2013) 'Recommender systems survey', *Knowledge-Based Systems*. doi: 10.1016/j.knosys.2013.03.012.

Bokde, D., Girase, S. and Mukhopadhyay, D. (2015) 'Matrix Factorization model in Collaborative Filtering algorithms: A survey', in *Procedia Computer Science*. doi: 10.1016/j.procs.2015.04.237.

Bolchini, C., Curino, C., Schreiber, F. A. and Tanca, L. (2006) 'Context integration for mobile data tailoring', in *SEBD 2006 - Proceedings of the 14th Italian Symposium on Advanced Databases Systems*. doi: 10.1109/mdm.2006.52.

Braunhofer, M., Elahi, M. and Ricci, F. (2014) 'STS: A context-aware mobile recommender system for places of interest', in *CEUR Workshop Proceedings*.

Cambria, E. (2016) ‘Affective Computing and Sentiment Analysis’, *IEEE Intelligent Systems*, 31(2), pp. 102–107. doi: 10.1109/MIS.2016.31.

Casillo, M., Clarizia, F., Colace, F., Lombardi, M., Pascale, F. and Santaniello, D. (2019) ‘An approach for recommending contextualized services in e-tourism’, *Information (Switzerland)*. doi: 10.3390/INFO10050180.

Casillo, M., Clarizia, F., D’Aniello, G., De Santo, M., Lombardi, M. and Santaniello, D. (2020) ‘CHAT-Bot: A cultural heritage aware teller-bot for supporting touristic experiences’, *Pattern Recognition Letters*. doi: 10.1016/j.patrec.2020.01.003.

Casillo, M., Colace, F., Santo, M. De, Lemma, S. and Lombardi, M. (2019) ‘CAT: a context aware teller for supporting tourist experiences’, *International Journal of Computational Science and Engineering*, 20(1), p. 69. doi: 10.1504/IJCSE.2019.103255.

Castiglione, A., Colace, F., Moscato, V. and Palmieri, F. (2018) ‘CHIS: A big data infrastructure to manage digital cultural items’, *Future Generation Computer Systems*. doi: 10.1016/j.future.2017.04.006.

Cerezo, J., Kubelka, J., Robbes, R. and Bergel, A. (2019) ‘Building an expert recommender chatbot’, in *Proceedings - 2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering, BotSE 2019*. doi: 10.1109/BotSE.2019.00022.

Chang, S. K., Yung, D., Colace, F., Greco, L., Lemma, S. and Lombardi, M. (2015) ‘An adaptive contextual recommender system: A slow intelligence perspective’, in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*. doi: 10.18293/SEKE2015-080.

Chen, G. and Kotz, D. (2000) ‘A survey of context-aware mobile computing research [Un estudio de la investigación sobre computación móvil sensible al contexto]’, *Time*.

Chen, Z., Zhao, W. and Wang, S. (2021) ‘Kernel meets recommender systems: A multi-kernel interpolation for matrix completion’, *Expert Systems with Applications*, 168(114436), pp. 1–10. doi: 10.1016/j.eswa.2020.114436.

Chianese, A. and Piccialli, F. (2016) ‘A smart system to manage the context evolution in the Cultural Heritage domain’, *Computers and Electrical Engineering*. doi: 10.1016/j.compeleceng.2016.02.008.

Clarizia, F., Colace, F., Lombardi, M., Pascale, F. and Santaniello, D. (2018) ‘Chatbot: An education support system for student’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/978-3-030-01689-0_23.

Colace, F., Lemma, S., Lombardi, M., De Santo, M. and Casillo, M. (2017) ‘A Context-Aware Mobile Solution for Assisting Tourists in a Smart Environment’, in *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*. doi: 10.24251/hicss.2017.061.

Colace, Francesco, Moscato, V., Quintarelli, E., Rabosio, E. and Tanca, L. (2015) ‘Context Awareness in Pervasive Information Management’, in, pp. 235–256. doi: 10.1007/978-3-319-20062-0_12.

Colace, F., De Santo, M., Greco, L., Amato, F., Moscato, V., Persia, F. and Picariello, A. (2015) ‘A user-centered approach for social recommendations’, in *ACHI 2015 - 8th International Conference on Advances in Computer-Human Interactions*.

Colace, F., De Santo, M., Greco, L., Amato, F., Moscato, V. and Picariello, A. (2014) ‘Terminological ontology learning and population using latent dirichlet allocation’, *Journal of Visual Languages and Computing*. doi: 10.1016/j.jvlc.2014.11.001.

Colace, F., De Santo, M., Greco, L., Lemma, S., Lombardi, M., Moscato, V. and Picariello, A. (2014) ‘A context-aware framework for cultural heritage applications’, in *Proceedings - 10th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2014*. doi: 10.1109/SITIS.2014.14.

Colace, Francesco, De Santo, M., Greco, L., Moscato, V. and Picariello, A. (2015) ‘A collaborative user-centered framework for recommending items in Online Social Networks’, *Computers in Human Behavior*. doi: 10.1016/j.chb.2014.12.011.

Colace, F., Santo, M. De, Lemma, S., Lombardi, M., Rossi, A., Santoriello, A., Terribile, A. and Vigorito, M. (2016) ‘How to Describe Cultural Heritage Resources in the Web 2.0 Era?’, in *Proceedings - 11th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2015*. doi: 10.1109/SITIS.2015.50.

Colace, Francesco, De Santo, M., Moscato, V., Picariello, A., Schreiber, F. A. and Tanca, L. (2015a) ‘PATCH: A Portable Context-Aware ATlas for Browsing Cultural Heritage’, in, pp. 345–361. doi: 10.1007/978-3-319-20062-0_17.

Colace, Francesco, De Santo, M., Moscato, V., Picariello, A., Schreiber, F. A. and Tanca, L. (2015b) ‘Pervasive Systems Architecture and the Main Related Technologies’, in. doi: 10.1007/978-3-319-20062-0_2.

Colace, F., De Santo, M., Pascale, F., Lemma, S. and Lombardi, M. (2017) ‘BotWheels: A petri net based Chatbot for recommending tires’, in *DATA 2017 - Proceedings of the 6th International Conference on Data Science,* *Technology and Applications*. doi: 10.5220/0006491903500358.

Colombo-Mendoza, L. O., Valencia-García, R., Rodríguez-González, A., Alor-Hernández, G. and Samper-Zapater, J. J. (2015) 'RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes', *Expert Systems with Applications*. doi: 10.1016/j.eswa.2014.09.016.

Conte, D. (2020) 'Dynamical low-rank approximation to the solution of parabolic differential equations', *Applied Numerical Mathematics*. doi: 10.1016/j.apnum.2020.05.011.

Conte, D. and Lubich, C. (2010) 'An error analysis of the multi-configuration time-dependent Hartree method of quantum dynamics', *ESAIM: Mathematical Modelling and Numerical Analysis*. doi: 10.1051/m2an/2010018.

Coutaz, J., Crowley, J. L., Dobson, S. and Garlan, D. (2005) 'Context is key', *Communications of the ACM*. doi: 10.1145/1047671.1047703.

D'aniello, G., Gaeta, M. and Reformat, M. Z. (2017) 'Collective perception in smart tourism destinations with rough sets', in *2017 3rd IEEE International Conference on Cybernetics, CYBCONF 2017 - Proceedings*. doi: 10.1109/CYBConf.2017.7985765.

Desrosiers, C. and Karypis, G. (2011) 'A Comprehensive Survey of Neighborhood-based Recommendation Methods', in *Recommender Systems Handbook*. doi: 10.1007/978-0-387-85820-3_4.

Dey, A. K. (2001) 'Understanding and using context', *Personal and Ubiquitous Computing*. doi: 10.1007/s007790170019.

Dey, A. K., Sohn, T., Streng, S. and Kodama, J. (2006) 'iCAP: Interactive prototyping of context-aware applications', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/11748625_16.

Dominici, M., Pietropaoli, B. and Weis, F. (2012) 'Experiences in managing uncertainty and ignorance in a lightly instrumented smart home', *International Journal of Pervasive Computing and Communications*. doi: 10.1108/17427371211262635.

Eirinaki, M., Gao, J., Varlamis, I. and Tserpes, K. (2018) 'Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions', *Future Generation Computer Systems*. doi: 10.1016/j.future.2017.09.015.

Fink, J., Kobsa, A. and Nill, A. (1998) 'Adaptable and adaptive information provision for all users, including disabled and elderly people', *New Review* *of Hypermedia and Multimedia*. doi: 10.1080/13614569808914700.

Gabriel, Y. and Connell, N. A. D. C. (2010) ‘Co-creating stories: Collaborative experiments in storytelling’, *Management Learning*. doi: 10.1177/1350507609358158.

Gene H. Golub (2013) *Matrix Computations,4th Edition*, *The Johns Hopkins University Press*.

Geuens, S., Coussement, K. and De Bock, K. W. (2018) ‘A framework for configuring collaborative filtering-based recommendations derived from purchase data’, *European Journal of Operational Research*. doi: 10.1016/j.ejor.2017.07.005.

Gluhak, A. and Schott, W. (2007) ‘A WSN system architecture to capture context information for beyond 3G communication systems’, in *Proceedings of the 2007 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP*. doi: 10.1109/ISSNIP.2007.4496818.

Gunawardana, A. and Shani, G. (2009) ‘A survey of accuracy evaluation metrics of recommendation tasks’, *Journal of Machine Learning Research*.

Haller, E. and Rebedea, T. (2013) ‘Designing a chat-bot that simulates an historical figure’, in *Proceedings - 19th International Conference on Control Systems and Computer Science, CSCS 2013*. doi: 10.1109/CSCS.2013.85.

Hand, D. J. and Till, R. J. (2001) ‘A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems’, *Machine Learning*. doi: 10.1023/A:1010920819831.

He, X., He, Z., Song, J., Liu, Z., Jiang, Y. G. and Chua, T. S. (2018) ‘NAIS: Neural Attentive Item Similarity Model for Recommendation’, *IEEE Transactions on Knowledge and Data Engineering*. doi: 10.1109/TKDE.2018.2831682.

Hong, J., Suh, E. H., Kim, J. and Kim, S. Y. (2009) ‘Context-aware system for proactive personalized service based on context history’, *Expert Systems with Applications*. doi: 10.1016/j.eswa.2008.09.002.

Ilarri, S., Trillo-Lado, R. and Hermoso, R. (2018) ‘Datasets for context-aware recommender systems: Current context and possible directions’, in *Proceedings - IEEE 34th International Conference on Data Engineering Workshops, ICDEW 2018*. doi: 10.1109/ICDEW.2018.00011.

Karabadji, N. E. I., Beldjoudi, S., Seridi, H., Aridhi, S. and Dhifli, W. (2018) ‘Improving memory-based user collaborative filtering with evolutionary multi-objective optimization’, *Expert Systems with Applications*. doi:

10.1016/j.eswa.2018.01.015.

Karatzoglou, A., Amatriain, X., Baltrunas, L. and Oliver, N. (2010) 'Multiverse Recommendation: N-dimensional Tensor Factorization for context-aware Collaborative Filtering', in *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems*. doi: 10.1145/1864708.1864727.

Kickmeier-Rust, M. D., Gbel, S. and Albert, D. (2008) '80Days: Melding adaptive educational technology and adaptive and interactive storytelling in digital educational games', in *CEUR Workshop Proceedings*.

Kohavi, R., Henne, R. M. and Sommerfield, D. (2007) 'Practical guide to controlled experiments on the web: Listen to your customers not to the hippo', in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. doi: 10.1145/1281192.1281295.

Kolda, T. G. and Bader, B. W. (2009) 'Tensor decompositions and applications', *SIAM Review*. doi: 10.1137/07070111X.

Koren, Y. (2010) 'Collaborative filtering with temporal dynamics', *Communications of the ACM*. doi: 10.1145/1721654.1721677.

Koren, Y., Bell, R. and Volinsky, C. (2009) 'Matrix factorization techniques for recommender systems', *Computer*. doi: 10.1109/MC.2009.263.

Kosir, A. O. and M. T. and M. K. and T. P. and J. T. and A. (2012) 'LDOS-CoMoDa dataset'.

Košir, A., Odić, A., Kunaver, M., Tkalčič, M. and Tasič, J. F. (2011) 'Database for contextual personalization', *Elektrotehniski Vestnik/Electrotechnical Review*.

Lambert, J. (2007) 'Digital storytelling', *Futurist*. doi: 10.4135/9781452230634.n41.

De Lathauwer, L., De Moor, B. and Vandewalle, J. (2000) 'A multilinear singular value decomposition', *SIAM Journal on Matrix Analysis and Applications*. doi: 10.1137/S0895479896305696.

Lee, P., Lakshmanan, L. V. S. and Milios, E. (2014) 'CAST: A context-aware story-teller for streaming social content', in *CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management*. doi: 10.1145/2661829.2661859.

Liu, J., Wu, C. and Liu, W. (2013) 'Bayesian probabilistic matrix factorization with social relations and item contents for recommendation', *Decision Support Systems*. doi: 10.1016/j.dss.2013.04.002.

Lord, N., Golub, G. H. and Loan, C. F. Van (1999) 'Matrix Computations', *The Mathematical Gazette*. doi: 10.2307/3621013.

Missaoui, S., Kassem, F., Viviani, M., Agostini, A., Faiz, R. and Pasi, G. (2019) 'LOOKER: a mobile, personalized recommender system in the tourism domain based on social media user-generated content', *Personal and Ubiquitous Computing*. doi: 10.1007/s00779-018-01194-w.

Mnih, A. and Salakhutdinov, R. (2007) 'Probabilistic matrix factorization', *Advances in neural information processing systems*.

Napoletano, P., Boccignone, G. and Tisato, F. (2015) 'Attentive Monitoring of Multiple Video Streams Driven by a Bayesian Foraging Strategy', *IEEE Transactions on Image Processing*. doi: 10.1109/TIP.2015.2431438.

Oberlander, J., O'Donnell, M., Mellish, C. and Knott, A. (1998) 'Conversation in the museum: Experiments in dynamic hypermedia with the intelligent labelling explorer', *New Review of Hypermedia and Multimedia*. doi: 10.1080/13614569808914693.

Oku, K., Nakajima, S., Miyazaki, J. and Uemura, S. (2006) 'Context-aware SVM for context-dependent information recommendation', in *Proceedings - IEEE International Conference on Mobile Data Management*. doi: 10.1109/MDM.2006.56.

Orsi, G. and Tanca, L. (2011) 'Context Modelling and Context-Aware Querying', in. doi: 10.1007/978-3-642-24206-9_13.

Page, L., Brin, S., Motwani, R. and Winograd, T. (1998) 'The PageRank Citation Ranking: Bringing Order to the Web', *World Wide Web Internet And Web Information Systems*. doi: 10.1.1.31.1768.

Paradarami, T. K., Bastian, N. D. and Wightman, J. L. (2017) 'A hybrid recommender system using artificial neural networks', *Expert Systems with Applications*. doi: 10.1016/j.eswa.2017.04.046.

Pascoe, J., Ryan, N. and Morse, D. (1999) 'Issues in developing context-aware computing', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/3-540-48157-5_20.

Pearce, C. (1994) 'The ins & outs of non-linear storytelling', *ACM SIGGRAPH Computer Graphics*. doi: 10.1145/178951.178956.

Perera, C., Zaslavsky, A., Christen, P. and Georgakopoulos, D. (2014) 'Context aware computing for the internet of things: A survey', *IEEE Communications Surveys and Tutorials*. doi: 10.1109/SURV.2013.042313.00197.

Petrelli, D., Not, E., Zancanaro, M., Strapparava, C. and Stock, O. (2001)

‘Modelling and adapting to context’, *Personal and Ubiquitous Computing*. doi: 10.1007/s007790170023.

Piccialli, F. and Chianese, A. (2017) ‘The Internet of Things Supporting Context-Aware Computing: A Cultural Heritage Case Study’, *Mobile Networks and Applications*. doi: 10.1007/s11036-017-0810-4.

Pittarello, F. (2006) ‘A simple story: Using an agents’ based context-aware architecture for storytelling’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/11944577_3.

Ponte, J. M. and Croft, W. B. (1998) ‘Language modeling approach to information retrieval’, *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*. doi: 10.1145/290941.291008.

Portugal, I., Alencar, P. and Cowan, D. (2018) ‘The use of machine learning algorithms in recommender systems: A systematic review’, *Expert Systems with Applications*. doi: 10.1016/j.eswa.2017.12.020.

Ramirez-Garcia, X. and García-Valdez, M. (2014) ‘Post-filtering for a restaurant context-aware recommender system’, *Studies in Computational Intelligence*. doi: 10.1007/978-3-319-05170-3_49.

Ranjbar Kermany, N. and Alizadeh, S. H. (2017) ‘A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques’, *Electronic Commerce Research and Applications*. doi: 10.1016/j.elerap.2016.12.005.

Rauseo, A., Martinenghi, D. and Tanca, L. (2013) ‘Contextual data tailoring using ASP’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/978-3-642-36008-4-5.

Raverdy, P. G., Riva, O., De La Chapelle, A., Chibout, R. and Issarny, V. (2006) ‘Efficient context-aware service discovery in multi-protocol pervasive environments’, in *Proceedings - IEEE International Conference on Mobile Data Management*. doi: 10.1109/MDM.2006.78.

Raza, S. and Ding, C. (2019) ‘Progress in context-aware recommender systems - An overview’, *Computer Science Review*. doi: 10.1016/j.cosrev.2019.01.001.

Ricci, F., Shapira, B. and Rokach, L. (2015) ‘Recommender systems: Introduction and challenges’, in *Recommender Systems Handbook, Second Edition*. doi: 10.1007/978-1-4899-7637-6_1.

Van Rijsbergen, C. J. (1979) ‘Information Retrieval, 2nd edition’, *Butterworths*.

Robin, B. (2006) 'The educational uses of digital storytelling', ... *for Information Technology & Teacher Education ...*.

Robin, B. R. (2008) 'Digital storytelling: A powerful technology tool for the 21st century classroom', *Theory into Practice*. doi: 10.1080/00405840802153916.

Rodríguez, J. D., Pérez, A. and Lozano, J. A. (2010) 'Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*. doi: 10.1109/TPAMI.2009.187.

Ruotsalo, T., Haav, K., Stoyanov, A., Roche, S., Fani, E., Deliai, R., Mäkelä, E., Kauppinen, T. and Hyvönen, E. (2013) 'SMARTMUSEUM: A mobile recommender system for the Web of Data', *Journal of Web Semantics*. doi: 10.1016/j.websem.2013.03.001.

Said, A., De Luca, E. W. and Albayrak, S. (2011) 'Inferring contextual user profiles improving recommender performance', in *CEUR Workshop Proceedings*.

Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M. and Oliveira, A. (2011) 'Smart cities and the future internet: Towards cooperation frameworks for open innovation', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/978-3-642-20898-0_31.

Schilit, B., Adams, N. and Want, R. (1995) 'Context-aware computing applications', in *Mobile Computing Systems and Applications - Workshop Proceedings*. doi: 10.1109/wmcsa.1994.16.

Schmidt, A., Beigl, M. and Gellersen, H. W. (1999) 'There is more to context than location', *Computers and Graphics (Pergamon)*. doi: 10.1016/S0097-8493(99)00120-X.

Shani, G. and Gunawardana, A. (2011) 'Evaluating Recommendation Systems', in *Recommender Systems Handbook*. doi: 10.1007/978-0-387-85820-3_8.

Shen, X., Tan, B. and Zhai, C. (2005) 'Context-sensitive information retrieval using implicit feedback', in *SIGIR 2005 - Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. doi: 10.1145/1076034.1076045.

Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M. and Hanjalic, A. (2014) 'CARS2: Learning context-aware representations for context-aware recommendations', in *CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management*. doi: 10.1145/2661829.2662070.

Shin, D., Lee, J. W., Yeon, J. and Lee, S. G. (2009) ‘Context-aware recommendation by aggregating user context’, in *2009 IEEE Conference on Commerce and Enterprise Computing, CEC 2009*. doi: 10.1109/CEC.2009.38.

Son, J. and Kim, S. B. (2018) ‘Academic paper recommender system using multilevel simultaneous citation networks’, *Decision Support Systems*. doi: 10.1016/j.dss.2017.10.011.

Spaniol, M., Klamma, R., Sharda, N. and Jarke, M. (2006) ‘Web-based learning with non-linear multimedia stories’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/11925293_23.

Stock, O. (1993) ‘ALFRESCO: Enjoying The Combination of NLP and Hypermedia for Information Exploration’, *In AAAI Workshop on Intelligent Multimedia Interfaces*.

Sun, Y. and Zhang, Y. (2018) ‘Conversational recommender system’, in *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*. doi: 10.1145/3209978.3210002.

Symeonidis, P. and Zioupos, A. (2016) *Matrix and Tensor Factorization Techniques for Recommender Systems, Book*.

Theimer, M. M. and Schilit, B. N. (1994) ‘Disseminating Active Map Information to Mobile Hosts’, *IEEE Network*. doi: 10.1109/65.313011.

Tiwari, S. and Kaushik, S. (2017) ‘Evolving recommendations from past travel sequences using soft computing techniques’, in *International Journal of Computational Science and Engineering*. doi: 10.1504/IJCSE.2017.084156.

Villegas, N. M., Sánchez, C., Díaz-Cely, J. and Tamura, G. (2018) ‘Characterizing context-aware recommender systems: A systematic literature review’, *Knowledge-Based Systems*. doi: 10.1016/j.knosys.2017.11.003.

De Virgilio, R., Torlone, R. and Houben, G. J. (2006) ‘A rule-based approach to content delivery adaptation in web information systems’, in *Proceedings - IEEE International Conference on Mobile Data Management*. doi: 10.1109/MDM.2006.16.

Wang, C. D., Deng, Z. H., Lai, J. H. and Yu, P. S. (2019) ‘Serendipitous recommendation in e-commerce using innovator-based collaborative filtering’, *IEEE Transactions on Cybernetics*. doi: 10.1109/TCYB.2018.2841924.

Wang, D., Liang, Y., Xu, D., Feng, X. and Guan, R. (2018) ‘A content-based recommender system for computer science publications', *Knowledge-Based Systems*. doi: 10.1016/j.knosys.2018.05.001.

Wang, R., Cheng, H. K., Jiang, Y. and Lou, J. (2019) 'A novel matrix factorization model for recommendation with LOD-based semantic similarity measure', *Expert Systems with Applications*. doi: 10.1016/j.eswa.2019.01.036.

Wang, Y., Wang, P., Liu, Z. and Zhang, L. Y. (2021) 'A new item similarity based on α-divergence for collaborative filtering in sparse data', *Expert Systems with Applications*, 166(114074), p. 114074. doi: 10.1016/j.eswa.2020.114074.

Xu, Z., Chen, L. and Chen, G. (2015) 'Topic based context-aware travel recommendation method exploiting geotagged photos', *Neurocomputing*. doi: 10.1016/j.neucom.2014.12.043.

Yao, Y. Y. (1995) 'Measuring retrieval effectiveness based on user preference of documents', *Journal of the American Society for Information Science*. doi: 10.1002/(SICI)1097-4571(199503)46:2<133::AID-ASI6>3.0.CO;2-Z.

Yassine, A., Mohamed, L. and Al Achhab, M. (2021) 'Intelligent recommender system based on unsupervised machine learning and demographic attributes', *Simulation Modelling Practice and Theory*. doi: 10.1016/j.simpat.2020.102198.

Ye, J. and Papavassiliou, S. (2008) 'An analytical framework for the modelling and evaluation of the mobile agent based distributed network management paradigm', *International Journal of High Performance Computing and Networking*. doi: 10.1504/IJHPCN.2008.022303.

Zalake, N. and Naik, G. (2019) 'Generative Chat Bot Implementation Using Deep Recurrent Neural Networks and Natural Language Understanding', *SSRN Electronic Journal*. doi: 10.2139/ssrn.3362123.

Zhai, C. and Lafferty, J. (2004) 'A study of smoothing methods for language models applied to information retrieval', *ACM Transactions on Information Systems*. doi: 10.1145/984321.984322.

Zhang, F., Yuan, N. J., Lian, D., Xie, X. and Ma, W. Y. (2016) 'Collaborative knowledge base embedding for recommender systems', in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. doi: 10.1145/2939672.2939673.

Zhao, J., Geng, X., Zhou, J., Sun, Q., Xiao, Y., Zhang, Z. and Fu, Z. (2019) 'Attribute mapping and autoencoder neural network based matrix factorization initialization for recommendation systems', *Knowledge-Based Systems*. doi: 10.1016/j.knosys.2018.12.022.

Zheng, Y., Burke, R. and Mobasher, B. (2012) 'Differential context relaxation for context-aware travel recommendation', in *Lecture Notes in Business Information Processing*. doi: 10.1007/978-3-642-32273-0_8.

Zheng, Y., Burke, R. and Mobasher, B. (2013) 'The role of emotions in context-aware recommendation', in *CEUR Workshop Proceedings*.

Zheng, Y., Mobasher, B. and Burke, R. (2014) 'Context recommendation using multi-label classification', in *Proceedings - 2014 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2014*. doi: 10.1109/WI-IAT.2014.110.

Zheng, Y., Mobasher, B. and Burke, R. (2016) 'CARSKit: A Java-Based Context-Aware Recommendation Engine', in *Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015*. doi: 10.1109/ICDMW.2015.222.

Zimmermann, A., Lorenz, A. and Oppermann, R. (2007) 'An operational definition of context', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi: 10.1007/978-3-540-74255-5_42.

Appendix A

A.1 The Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) is used to reduce a rectangular matrix in a diagonal form by a suitable pre-multiplication and post-multiplication by orthogonal matrices. It is recalled that an orthogonal matrix is a square invertible matrix whose inverse coincides with its transpose and that a square matrix $A \in R^{n\times n}$ is similar to a matrix $B \in R^{n\times n}$ if there exists a square invertible matrix $S \in R^{n\times n}$, such that $S^{-1}AS = B$.

Theorem A.1: *Let $m, n \in N$ and let $R \in R^{m\times n}$, then $\exists U \in R^{m\times m}$ and $\exists V \in R^{n\times n}$ orthogonal matrices such that:*

$$U^T RV = D = diag(\sigma_1, \ldots, \sigma_p) \in R^{m\times n}, \quad p = \min\{m, n\} \qquad (A.1)$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > \sigma_{k+1} = \cdots = \sigma_p = 0 \qquad (A.2)$$

From $(A.1)$, it quickly follows:

$$R = UDV^T \qquad (A.3)$$

The elements of the matrices obtained by singular value factorization are defined as follows:

- The columns of the matrix $U = (u_1, \ldots, u_m) \in R^{m\times m}$ are defined to be the left singular vectors.

- The columns of the matrix $V = (v_1, \ldots, v_n) \in R^{n\times n}$ are defined to be the right singular vectors.

- $D = diag(\sigma_1, \ldots, \sigma_p) \in R^{m\times n}$, $p = min\{m, n\}$ is the singular value matrix.

In particular, it can easily be proved that the columns of the matrix U are the eigenvectors of the square matrix $RR^T \in R^{m\times m}$ and, similarly, that the columns of the matrix V are the eigenvectors of the matrix R^TR $\in R^{n\times n}$. The non-zero elements of the diagonal matrix D are the square roots of the eigenvalues of the matrices RR^T and R^TR.

It is useful to find low-rank approximations of a given matrix R in the computational context to know the associated approximation error. For this purpose, the *Eckart - Young Theorem* is stated (Lord *et al.*, 1999).

Theorem A.2 (Eckart-Young): *Let $R \in R^{m\times n}$ and U, V, D be given by Theorem 2.1. Let $r = r(R)$ the rank of the given matrix, with $r \leq p = min\{m, n\}$. We define:*

$$R_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T = U_k D_k V_k^T \qquad (A.4)$$

with $k \leq r$ and $U_k = (u_1, \dots, u_k) \in R^{m\times k}$ matrix of the left singular vectors without the last $m - k$ columns, $V_k = (v_1, \dots, v_k) \in R^{n\times k}$ matrix of the right singular vectors without the last $n - k$ columns, $D_k = diag(\sigma_1, \dots, \sigma_k) \in R^{k\times k}$ matrix of singular values without the last $m - k$ lines and $n - k$ columns.

Then:

$$\min_{r(B)=k} \|R - B\|_F^2 = \|R - R_k\|_F^2 = \sum_{i=k+1}^{p} \sigma_i^2 \qquad (A.5)$$

$$\min_{r(B)=k} \|R - B\|_2 = \|R - R_k\|_2 = \sigma_{k+1} \qquad (A.6)$$

where $\|\cdot\|_F$ and $\|\cdot\|_2$ are the Frobenius and the spectral norm, respectively.

Theorem 2.2 establishes that, considering only the first $k \leq r(R)$ singular values of the matrix R we obtain an approximation of the matrix R using a low-rank matrix R_k having rank k. The associated error is estimated by the relations $(A.5)$ and $(A.6)$. In this way, it is possible to reduce the computational cost relating to singular value factorization to estimate the acceptable error based on the relationship $(A.2)$. The low-rank approximation of matrices is used in many applications such as control theory, signal processing, machine learning, image compression, information retrieval, quantum physics (Conte and Lubich, 2010; Conte, 2020). In many of these applications, the matrices are not constant, but they vary with the time, thus leading to the need for dynamical low-rank approximation.

A.1.1 The use of SVD in Collaborative Filtering methods

Average Rating Filling (ARF), Stochastic Gradient Descent (SGD) and Biased Stochastic Gradient Descent (BSGD) methods are presented below. Each of them peculiarly uses the singular value factorization to predict the unknown ratings when some ratings are known. In particular, SGD and BSGD use the singular value decomposition in order to define numerical user and item features through the latent factors of the rating matrix. The number k of selected latent factors is defined as a priori and acts as a dimension of the vectors that describe the user and item attributes.

A.1.1.1 Average Rating Filling

Let $R \in R^{m\times n}$ be the matrix of known ratings, the Average Rating Filling method replaces the null elements of the R matrix, representing the unknown ratings, with the column averages of the same. If there is a column without known ratings, and therefore only null elements, it will be replaced by the column containing the average of all known ratings contained in R. The constructed matrix is denoted by $\tilde{R}$.

At the end of the replacement phase for the null elements, the reduced SVD for the matrix $\tilde{R}$ is used, obtaining the approximation $\widetilde{R_k}$ through the matrices U_k, V_k and D_k according to *Theorem of Eckart-Young*.

At this point, the matrices P and Q are obtained according to the relations:

$$P = U_k\sqrt{D_k} \in R^{m\times k} \qquad (A.7)$$

$$Q = \sqrt{D_k}V_k^T \in R^{k\times n} \qquad (A.8)$$

The rating forecasts are obtained through the product of the obtained matrices P and Q, and are inserted in the matrix $\hat{R}$.

A.1.1.2 Stochastic Gradient Descent

Let some elements of the rating matrix R be known. The purpose of the Stochastic Gradient Descent method is to create vectors $p_i, i = 1, \dots, m$ associated with the characteristics of users and vectors $q_j, j = 1, \dots, n$ associated with the items features (R. Wang *et al.*, 2019; Zhao *et al.*, 2019). These vectors are obtained through the introduction of random k - dimensional vectors, which are updated through the rule:

$$\theta \leftarrow \theta - \alpha \frac{\partial f}{\partial \theta} \qquad (A.9)$$

where f denotes the mean square error function.

Let $N = \{(i,j) : r_{ij} \neq 0\}$ be the set of pairs of which the rating is known and $\widehat{r_{ij}} =< p_i, q_j >$ the rating prediction provided by the procedure. The error function f is defined as follows:

$$f(p_1, \dots, p_m, q_1, \dots, q_n) = \sum_{(i,j) \in N} \left(r_{ij} - < p_i, q_j >\right)^2 = \sum_{(i,j) \in N} f_{ij}\left(p_i, q_j\right) \; (A.10)$$

with $f_{ij}\left(p_i, q_j\right) = \left(r_{ij} - < p_i, q_j >\right)^2$.

From the function f_{ij} the components of the associated gradient are obtained:

$$\frac{\partial f_{ij}}{\partial p_i} = -2q_j\left(r_{ij} - < p_i, q_j >\right) \quad \forall i = 1, \dots, m \qquad (A.11)$$

$$\frac{\partial f_{ij}}{\partial q_j} = -2p_i\left(r_{ij} - < p_i, q_j >\right) \quad \forall j = 1, \dots, n \qquad (A.12)$$

Through the gradient formulas, the learning rules are obtained:

$$error = r_{ij} \prec p_i, q_j > \qquad (A.13)$$

$$p_i \leftarrow p_i + \alpha \cdot error \cdot q_j \qquad (A.14)$$

$$q_j \leftarrow q_j + \alpha \cdot error \cdot p_i \qquad (A.15)$$

With α the learning factor has been denoted, a constant that must be determined a priori.

A.1.1.3 Biased Stochastic Gradient Descent

Koren subsequently modified the Stochastic Gradient Descent method (Zhao *et al.*, 2019) through a different way of calculating the rating predictions and a modified error function (Koren *et al.*, 2009).

Let μ be the average of known ratings, b_i the bias of the i - th user and b'_j the bias of the j - th item. The prediction of the rating associated with i - th user and j - th item is calculated using the formula:

$$\widehat{r_{ij}} = \mu + b_i + b'_j + < p_i, q_j > \qquad (A.16)$$

Let N be the set of the pairs (i, j) for which the associated rating is known. The error function to be minimized considered by Koren is the regularized squared error:

$$f(p_1, \dots, p_m, q_1, \dots, q_n) = \sum_{(i,j)\in N} \left(\left(r_{ij} - \widehat{r_{ij}}\right)^2 + \lambda\left(\|p_i{}^2\| + \left\|q_j{}^2\right\| + b_i^2 + b_j'^2\right) \right) \qquad (A.17)$$

with λ regularization constant.

By reasoning analogously to the Stochastic Gradient Descent method, the learning rules are obtained:

$$error = r_{ij} - \widehat{r_{ij}} \qquad (A.18)$$

$$b_i \leftarrow b_i + \alpha(error - \lambda b_i) \qquad (A.19)$$

$$b_j' \leftarrow b_j' + \alpha\left(error - \lambda b_j'\right) \qquad (A.20)$$

$$p_i \leftarrow p_i + \alpha\left(error \cdot q_j - \lambda p_i\right) \qquad (A.21)$$

$$q_j \leftarrow q_j + \alpha\left(error \cdot p_i - \lambda q_j\right) \qquad (A.22)$$

Appendix B

B.1 The Latent Dirichlet Allocation (LDA)

According to the The Latent Dirichlet Allocation (LDA) model, distribution of terms for each topic i is represented as a multinomial distribution φ_i drawn from a symmetric Dirichlet distribution with parameter β:

$$p(\phi_i|\beta) = \frac{\Gamma(W\beta)}{[\Gamma\beta]^W} \prod_{v=1}^{W} \emptyset_{iv}^{\beta-1} \qquad (B.1)$$

The topic distribution for a document d is also represented as a multinomial distribution Θ_d drawn by a Dirichlet distribution with parameter α:

$$p(\theta_d|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \emptyset_{di}^{\alpha_i - 1} \qquad (B.2)$$

In this way, the topic z_{dn} for each index n token can be chosen from the distribution of the document topics as:

$$p(z_{dn} = i|\theta_d) = \theta_{di} \qquad (B.3)$$

Each token w is chosen from a multinomial distribution associated with the selected topic:

$$p(w_{dn} = v|z_{dn} = i, \phi_i) = \emptyset_{iv} \qquad (B.4)$$

LDA aims to find patterns of co-occurrence terms in order to identify consistent topics. If LDA is used to learn a topic i and $p(w = v|z = i)$ is high for a specific term v, then every document d that contains the term v has a high probability for the topic i.

It is possible to state that all the terms co-occurring with the term v are more likely generated by the topic i.