

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *05/07/2017* **par :**

IMEN AKERMI

A Hybrid Model for Context-Aware Proactive Recommendation

JURY

KARINE ZEITOUNI	Professeur, Université de Versailles et Saint Quentin	Présidente
AMEL BOUZEGHOUB	Professeur, Telecom SudParis	Rapporteur
FAIEZ GARGOURI	Professeur, Université de Sfax	Rapporteur
GUILLAUME CABANAC	Maître de conférences, Université Toulouse 3	Examinateur
MOHAND BOUGHANEM	Professeur, Université Toulouse 3	Directeur
RIM FAIZ	Professeur, Université de Carthage	Co-directrice

École doctorale et spécialité :
MITT : Image, Information, Hypermedia

Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)

Directeur(s) de Thèse :
Mohand Boughanem et *Rim Faiz*

Rapporteurs :
Amel Bouzeghoub et *Faiez Gargouri*

A Hybrid Model for Context-Aware Proactive Recommendation

Imen Akermi

Acknowledgments

My sincere and deep gratitude to my supervisors, Pr Mohand Boughanem and Pr Rim Faiz for introducing me to the area of research, and for their availability and their real interest in this work.
I thank them for providing guidance throughout my studies and for their valuable advice. Their encouragements have been of great value for me. I also thank them for reading carefully this dissertation allowing me to improve its content.
I would like to thank the reviewers of this thesis, Pr Amel Bouzeghoub, Professor at Telecom SudParis University and Pr Faiez Gargouri, Professor at the University of Sfax, for their careful and detailed reading of my manuscript as well as for their encouraging and constructive notes. I would also like to express my gratitude to Dr Guillaume Cabanac for having accepted to examine my work.
Special thanks to the members of the IRIS team.
In closing, I would like to thank my family and my friends who have been of great support throughout my studies.

Abstract

Just-In-Time recommender systems involve all systems able to provide recommendations tailored to the preferences and needs of users in order to help them access useful and interesting resources within a large data space. The user does not need to formulate a query, this latter is implicit and corresponds to the resources that match the user's interests at the right time. Our work falls within this framework and focuses on developing a proactive context-aware recommendation approach for mobile devices that covers many domains. It aims at recommending relevant items that match users' personal interests at the right time without waiting for the users to initiate any interaction. Indeed, the development of mobile devices equipped with persistent data connections, geolocation, cameras and wireless capabilities allows current context-aware recommender systems (CARS) to be highly contextualized and proactive. Nevertheless, this requires to know how to efficiently combine the context dimensions. Several dimensions of context, such as location, time, users activities, needs, resources, light, noise, movement, etc., have to be managed and represented which require a big amount of information and are time consuming. Besides, the incorporation of too many context dimensions generate complex context models. On the other hand, context models integrating few dimensions are unable to figure out the whole user context.

Therefore, we propose, in a first part, the modelling of a situational user profile and the definition of an aggregation frame for contextual dimensions combination within a proactive recommendation approach. Indeed, many of the actual contextualized systems focus on a particular domain (tourism, movie, news ...) and apply specific context dimensions

according to it. However, most of them depend almost on the same context combination which includes location, time and user preferences with a slight difference on how to approach this information. Thus, we can take advantage from the same context information without encumbering the user's mobile and recommend items related to different domains. The approach that we present integrates information related to a user gathered from his browsing tendency along with mobile technologies in order to proactively recommend relevant information to the user. Therefore, the recommendation process entails a context model that figures out what and when to recommend the relevant information (news, movies, a place to visit, a restaurant, ...) to the user.

We also extend, in a second part, a situation assessment approach in which we tackle the intrusiveness aspect within the recommendation process. Actually, it is no longer enough for a recommender system to determine what to recommend according to the user's needs but it also has to deal with the risk of disturbing the user during the recommendation process. The situation assessment approach makes use of the user's context and the several applications and sensors embedded within the user's mobile device in order to figure out the situations in which the user might reject recommendations. It is about balancing recommendations against intrusive interruptions. As a matter of fact, there are different factors and situations that make the user less open to recommendations. As we are working within the context of mobile devices, we consider that mobile applications functionalities, such as the camera, the keyboard, the agenda, etc., are good representatives of the user's interaction with his device since they reflect most of the activities that a user could use in a mobile device on a daily basis such as texting messages, chatting, tweeting, browsing or taking selfies and pictures. These mobile functionalities along with the user's context are tackled within the approach that we propose to assess intrusiveness. Indeed, thee works that tackled this aspect, approached it as a user modelling issue and considered that intrusiveness is limited to figuring out implicitly the user's preferences and related information and as it comes to the works that integrated intrusiveness into the recommendation process, they only relied on the user's activity depicted from his agenda

to assess if they can send a recommendation or not. The approach that we present detects intrusiveness within a proactive recommendation approach, not only in terms of the user's agenda activity but also including user's context with its several level of representation and other applications embedded in the user's mobile device besides the agenda.
Keywords: Context modelling, Context-aware recommendation, Proactive recommendation, Risk-aware recommendation.

Publications

Our contributions have already appeared in the following scientific publications:
International conferences

- Akermi, I., Boughanem, M. and Faiz, R., Just-In-Time Recommendation Approach Within A Mobile Context. Proceedings of 2016 IEEE/WIC/ACM International Conference on Web Intelligence, October 13-16, 2016 in Omaha, Nebraska, USA.
- Akermi, I., and Faiz, R., Context-Aware Proactive Recommendation Approach. Proceedings of the 7th International Conference on Computational Collective Intelligence (ICCCI 2015), Madrid, Spain, September 21-23, 2015. Springer International Publishing.
- Akermi, I. and Faiz, R., Using Social Networks for Mobile Proactive Recommendation. Proceedings of the 11th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA 2014), Doha, Qatar, 10-13 November 2014.

National conferences

- Akermi, I., Boughanem, M. and Faiz, R., Une Approche de recommandation proactive dans un environnement mobile. Proceedings of the 33rd INFORSID conference, Biarritz, France, May 26-29, 2015.

Contents

I Introduction **13**

1 Introduction **14**

1.1 Motivation 14

1.2 Research issues and contribution 16

1.3 Thesis outline 18

II Literature Review **20**

2 Context-Aware Framework **21**

2.1 Introduction 21

2.2 Context definition 21

2.3 Context modelling 22

2.3.1 Key-value models 22

2.3.2 Mark-up models 23

2.3.3 Object-oriented models 23

2.3.4 Logic-based models 23

2.3.5 Ontology-based models 24

2.4 Context acquisition 24

2.5 Context dimensions 25

2.5.1 The user's profile 28

2.5.1.1 The user's profile acquisition 29

2.5.1.2 The user's profile construction . . . 31
2.5.1.3 The user's profile evolution . . . 32
2.5.2 Location . . . 32
2.5.3 Time . . . 33
2.5.4 The user's activity . . . 34
2.6 From context-aware systems to situation-aware systems . . . 36
2.7 Conclusion . . . 38

3 Recommender Systems 39
3.1 Introduction . . . 39
3.2 Definition . . . 40
3.3 Recommendation approaches . . . 41
3.3.1 Content-based approaches . . . 42
3.3.1.1 Representation techniques . . . 43
3.3.1.2 Recommendation algorithms . . . 43
3.3.1.2.a Keywords-based recommendation . . . 43
3.3.1.2.b Semantic-based recommendation . . . 45
3.3.2 Collaborative-based approach . . . 46
3.3.2.1 Neighbourhood-based recommendation . . . 47
3.3.2.2 Machine learning based recommendation . . . 48
3.3.3 Hybrid approaches . . . 49
3.3.4 Recommender systems issues . . . 51
3.4 Recommender systems evaluation . . . 53
3.4.1 Offline, online evaluation and user studies . . . 53
3.4.1.1 Offline evaluation . . . 53
3.4.1.2 User studies . . . 54
3.4.1.3 Online experiments . . . 55
3.4.2 User/Business-oriented Evaluation . . . 55
3.4.3 Alternative quality measures . . . 55

3.5 Mobile, proactive and context-aware recommender systems 56
3.5.1 Spatio-Temporal based systems 57
3.5.2 The user's current or past behaviour based systems 58
3.5.3 Activity-centric systems 59
3.6 Non-intrusive Recommendation 60
3.6.1 Definition 60
3.6.2 Identifying Intrusiveness 60
3.6.2.1 Non-intrusiveness as implicit user profiling 61
3.6.2.2 Non-intrusiveness as non-disturbing recommendation . . . 62
3.7 Conclusion 64

III A Proactive and Non-Intrusive Recommendation Approach 65

4 A Mobile Situation-Aware Proactive Recommendation Approach 66
4.1 Introduction 66
4.2 Situation-based proactive recommendation 67
4.2.1 Context modelling and acquisition 68
4.2.1.1 The user's profile 68
4.2.1.2 Time 70
4.2.1.3 Location 70
4.2.2 Information extraction and recommendation 71
4.3 Experiments 75
4.3.1 Experimental framework 75
4.3.1.1 The TREC Contexual Suggestion Track 75
4.3.1.2 Evaluation metrics 77
4.3.2 Evaluation of our approach 79
4.3.2.1 Profiles building 79
4.3.2.2 Contexts processing 80

4.3.2.3 The profile/suggestions matching phase 82
4.3.3 Participants' runs 83
4.3.4 Results 85
4.4 Implementation : A Mobile Application for Proactive Recommendation . . 89
4.4.1 Host of the application 89
4.4.2 Application scenario 89
4.5 Conclusion 94

5 Non-Intrusive Recommendations in a Mobile Context 95
5.1 Introduction 95
5.2 Situation assessment for non-intrusive recommendations 96
5.2.1 Situation modelling 96
5.2.2 Retrieval 98
5.2.2.1 Time feature similarity computation 98
5.2.2.2 The user's activity similarity computation 99
5.2.3 Reuse 100
5.2.4 Revise 101
5.3 Experiments 102
5.3.1 User Study 102
5.3.2 Evaluation 104
5.3.2.1 Results 104
5.3.2.2 Analysis 105
5.4 Conclusion 108

6 Conclusion and Research Directions 109
6.1 Contribution summary 109
6.2 Perspectives 112

List of Figures

2.1 The multi-dimensional concept of context in IR (Tamine-Lechani et al. 2010) 26

3.1 Content-based approach process 42

4.1 A general overview of the proposed approach 67
4.2 Activities through social networks 69
4.3 Hourly app usage by category (Böhmer et al. 2011) 72
4.4 The Home and the Facebook fragments 90
4.5 Menu fragment 91
4.6 News fragment 92
4.7 Restaurants fragment 92
4.8 Notification example 93
4.9 Activating notifications 93

5.1 Activity partitioning scheme 100
5.2 The user study overview 103
5.3 The recommendation accuracy using the intrusiveness aspect 104
5.4 The users' behaviour regarding some activities (study conducted with 1500 smartphone users) 106
5.5 The notification acceptance rate according to the time of the day 107
5.6 The notification acceptance rate according to the day of the week 107

List of Tables

4.1 The user study inter-agreement 85
4.2 The users' study and the TREC evaluations scores matching 86
4.3 The five P@5 measures sorted by WGT 87
4.4 The five MRR@5 measures sorted by WGT 88

Part I

Introduction

Chapter 1

Introduction

1.1 Motivation

The access to relevant information, adapted to the user needs and profile, is a key issue in the current context characterized by a massive proliferation of heterogeneous information resources. Research is now heading towards adapting classical recommendation systems to issue information relevant to the user specific needs, context and preferences. This area of research called Context Aware Recommendation (CAR) knows nowadays a great interest. The actual CAR systems aim to combine a set of technologies and knowledge about the user context not only in order to deliver the most appropriate information to the user need but also to produce a synthesis of the information needed and recommend it to the user without having him to issue any query at just the right time. It is called:

- Zero-Query Search
- Proactive Recommendation

In practice, recommender systems (RS) consist of Web applications that provide users with lists of items. Such items may correspond to different types of data such as news (Das et al. 2007), restaurants (Burke 2007), music (Su et al. 2010), movies (Miller et al. 2003), books (Mooney and Roy 2000), jokes (Miyahara and Pazzani 2000), web pages

(Pitkow and Pirolli 1999), scientific articles (Pavlov et al. 2004), etc. To achieve such goal, a RS needs to accumulate data about users and available items. Indeed, the two basic entities that appear in any recommendation system are the user and the item (the song for music, ...) to recommend (Arnautu 2012). The input data for a recommendation system are usually gathered using ratings which express the user's opinions on the items. They are normally supplied explicitly by the user and follow a specific numerical scale (example *1-bad* to *5-excellent*). Ratings can also be implicitly collected from the user's purchase history, web logs, reading and listening habits. A RS might also focuses on the content data, which are based on a textual analysis of documents related to the items evaluated by the user. The features extracted from this analysis are used as inputs to the RS in order to infer a user profile (Vozalis and Margaritis 2003). Demographic data, which refer to information such as age, sex and education of users, can also be considered within the recommendation process. However, this type of data is generally difficult to obtain and is normally collected explicitly.

Another key aspect, in recommendation approaches, is the use of context which stands for factors such as location, time and the user's current activity that describes or infers the user's situation. Work in context-aware recommendation makes use of one or all of these dimensions to describe the user and integrate him forward in the various phases of the recommendation process: *the information need reformulation*, *the selection of information resources* and *the information relevance evaluation*. However, this requires an efficient modeling of the context dimensions. Indeed, as mentioned (Mizzaro and Vassena 2011), several dimensions of context, such as location, time, users activities, resources in the nearbies, movement, etc., have to be managed and represented which requires a big amount of information and are time consuming. On the other hand, context models integrating few dimensions are unable to figure out the whole user context. Besides, relying on user's explicit rating data as feedback for recommendation puts a certain burden on the users.

This thesis aims to bring out a context-aware proactive approach that integrates the modelling of a situational user profile and the definition of an aggregation frame for contextual dimensions combination. Our approach deals with the domain dependency and the proactivity issues by integrating the concept of user's situation and the same context information without encumbering the user's mobile device and recommend relevant items related to different domains at the right time without waiting for the user to initiate any interaction or query.

We also hypothesized about the different factors that make the user less open to recommendations. Indeed, despite the relevance of the personalized information delivered to the user, this latter may choose to reject recommendations in certain situations. This abstinence does not concern the recommended information itself, but it takes part in the situation the user may be in and during which the user does not want to be disturbed. Thus, it is important to include the risk of disturbing the user within the recommendation process.

1.2 Research issues and contribution

Many of the actual recommender systems encounter certain limitations regarding the recommendation process and the context factors acquisition, to mention the explicit user profiling where users are requested to express their interests and input keywords or tags which is, most of the time, inconvenient in a mobile environment since it entails extra efforts from the user such as tagging, searching, or clicking (Nguyen and Riedl 2013). Mobile systems can help keep track of user's activities, preference and location. Besides, in order to recommend items related to the user's interests, various approaches depend only on the user's past or actual behaviour history. By behaviour, we mean Web browsing history/clicks (Shmueli-Scheuer et al. 2010, Das et al. 2007); previous visiting behaviours for location based systems (Li et al. 2012a, Pu et al. 2012) and previous reading patters for news recommendation systems (Lee and Park 2007a, IJntema et al. 2010, Gershman et al.

2011, Arora and Shah 2011, Athalye 2013, Dumitrescu and Santini 2012). However, we cannot only rely on the user's past behaviour since it may not contain enough information about the user's interests. A user can browse a web page by chance or visit a particular place without having a real interest in that place. Other approaches considered recommendation from an activity centric angle. They depend on specific triggers to launch the recommendation process. The triggers may take the form of ongoing conversation or activity, such as text messages, phone calls (Popescu-Belis et al. 2011); opened web pages or documents (Dumais et al. 2004, Karkali et al. 2013, Prekop and Burnett 2003) and the social media activity of the user, such as the content of the user's tweet stream on Twitter[1] (De Francisci Morales et al. 2012, O'Banion et al. 2012, Phelan et al. 2011). Nevertheless, we cannot wait for the user to perform an activity to initiate the recommendation process. One can simply open a document to work on without being related to it in any way or have a conversation about an issue that he/she is not concerned with any recommendation about. Furthermore, a recommender system can no longer be seen as just a way to help a user choose from a set of resources that he/she does not have enough knowledge about, but it has to be able to provide the user with relevant information when it is most needed at the right time without waiting for the user to undertake an activity. There is also the domain dependency issue that needs to be addressed. Indeed, many of the actual contextualized systems are domain dependent (tourism, movies, news ...) and have specific context dimensions to apply according to the domain. However, most of them rely almost on the same context combination which includes location, time and user preferences with a slight difference on how to approach these dimensions. Therefore, many different services related to different domains can take advantage from the same context information without encumbering the user's device. In addition, with the rapid growth of mobile applications, the user is increasingly confronted with a lot of information and tends to reject the recommended items in some situations. It is no longer enough for a recommender system to determine what to recommend according to users' needs, but it

[1] https://twitter.com/

also has to deal with the risk of disturbing the user during the recommendation process. The works that tackled this aspect, approached it as a user modelling issue and considered that intrusiveness is limited to figuring out implicitly the user's preferences and related information. As it comes to the works that integrated intrusiveness into the recommendation process, they only relied on the user's activity depicted from his agenda to assess if they can send a recommendation or not.
Our approach deals with these challenging problems and contributes to the existing body of knowledge by entailing and combining the following characteristics :

- A Non-dependent domain system: we propose to cover various domains in the recommendation process.
- A Proactive system: we propose a proactive situation-aware recommender system that can help users deal with information overload problem efficiently by recommending the right item that matches users' personal interests at the right time without waiting for users to initiate any interaction
- A Social Networks based system: the genuine interests of the user provided by social networks would be of a great help for user profiling. Indeed, social networks provide a wealth of information about the user's interests. Besides, they play a double-edged role, within the approach we present, as a foundation for user profile modelling and as information generating resource.
- A Non-intrusive system: Our approach integrates a situation assessment phase in which we use mobile technologies along with context dimensions in order to figure out what are the different factors that make the user less open to recommendation.

1.3 Thesis outline

This thesis report is structured into two parts: the first part presents the broader context in which our work is entailed namely context-aware recommendation systems. The second

part describes our contribution.

We provide in the first part a presentation of the basic concepts related to context-aware and proactive recommender systems. This part entails two chapters :

- Chapter 2 introduces the context concept and the notions that are related to it.
- Chapter 3 presents the different techniques and approaches regarding recommendation and proactive recommendation systems.

The second part of this report addresses our contribution and is composed of two chapters:

- Chapter 4 describes the contextual dimensions combination framework for proactive context-aware recommendation.
- Chapter 5 exposes the situation assessment phase that we propose to tackle intrusiveness within the recommendation process.

Part II

Literature Review

Chapter 2

Context-Aware Framework

2.1 Introduction

The use of contextual information is very crucial to boost the performance of systems that falls within different research areas such as Information Retrieval and Recommender Systems. Indeed, the contextual information illustrated through different aspects and factors help to provide the most appropriate information to the user when it is most needed.

In this chapter, we present the basic concepts related to context and the various factors that it entails.

2.2 Context definition

The concept of context has been addressed in many works and has been defined through different aspects. The "*context-aware*" term was first introduced by Schilit and Theimer (1994) as "*location, identities of nearby people and objects, and changes to those objects*". The definitions that came after just added other characteristics that describe the context such as time and season (Brown et al. 1997), identity and environment (Ryan et al. 1999) and the emotional state of the user (Dey 1998).

The most commonly and widely used definition for context was presented by Abowd et al. (1999) as follows:

"*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user, and applications themselves.*"

From a general angle, context is defined as the set of cognitive and social factors as well as the goals and intentions of the user during an activity.

An attempt to distinguish between these concepts has been the subject of other studies (Allen 1997, Sonnenwald 1999, Cool 2001) that indicate that there is a broader background behind these aspects such as the cognitive, the social and the professional environment which cover situations related to factors such as location, time and the current application. This is the generic sense of context that has been widely explored (Lawrence 2000, Quiroga and Mostafa 2002, Ingwersen and Järvelin 2005, Bottraud et al. 2004). To sum up, we can define context as "*a set of dimensions that describe and/or infer user intentions and perception of relevance*".

2.3 Context modelling

Context modelling provide a formal representation of the contextual information as a unified structure (part of an ontology, term vectors set, a set of concepts,...) or as a set of information with different and specific structures. In the following sections, we describe the most frequent modelling approaches classified by (Strang and Linnhoff-Popien 2004).

2.3.1 Key-value models

They are based on a set of weighted keywords (or vectors of terms) represented by the Salton vector model (Salton and Yang 1973). The set of terms stands generally for the user's interests.

We can distinguish between set-representations that use a vector of weighted terms repre-

senting a particular interest (Lieberman 1995, Tamine et al. 2007) and those using classes of weighted terms vectors each of which stands for a category of interest (Gowan 2003, Sieg et al. 2004).

2.3.2 Mark-up models

These models are based on a hierarchical data structure using attributes, tags and content to model profiles (Musumba and Nyongesa 2013). Several descriptive models of context information are derived from this language to mention CC/PP[1] (Composite Capabilities/Preference Profile) and UAProf[2] (User Agent Profile).
These models provide a description of contextual factors including the basic constraints and the relationships.

2.3.3 Object-oriented models

This kind of context modelling is based on different object-oriented aspects such as encapsulation, inheritance and re-usability (Musumba and Nyongesa 2013). Hofer et al. (2003) addressed this modelling method and introduced the "HYDROGEN" approach. Each type of context used consists of several objects which also are the superclasses of several other elements of the context, including: time, network, location and user.

2.3.4 Logic-based models

It is a rule-based model where an inference process is applied in order to extract new facts from the existing rules. Then, the facts represents the contextual information in a formal way (Musumba and Nyongesa 2013).

[1]W3C. Composite Capabilities / Preferences Profile (CC/PP). http://www.w3.org/Mo- bile/CCPP
[2]WAPFORUM. User Agent Profile (UAProf). http://www.wapforum.org

2.3.5 Ontology-based models

They make use of domain ontologies or predefined concept hierarchies. The conceptual representation approach consists in specifying the levels of ontology concepts to consider, and then apply the data deployment process. Thus, the user context will be modelled as a conceptual network of nodes interconnected by respecting the topology of links defined in hierarchies or ontologies. There are different conceptual representations to consider: concepts hierarchy (Kim and Chan 2003), a portion of an ontology (Gauch et al. 2003, Sieg et al. 2007), concepts matrices (Liu et al. 2004) or concepts graphs (Daoud et al. 2009).

2.4 Context acquisition

Context is defined as a set of dimensions that cover situations related to factors such as location, time and the current activity. Work in context-aware applications makes use of one or all of these dimensions to describe the user and integrate him forward in the various phases of the recommendation process. Yeung (2011) defines context acquisition as "*the process of obtaining user context information*".
As stated by Adomavicius and Tuzhilin (2008), the contextual information can be gathered in different ways, namely:

- Explicitly: by asking users questions through web forms or before giving access to web pages or applications.
  Liu et al. (2004) ask the user to select explicitly concepts describing context from the ODP ontology[3] in order to identify the user's context.

- Implicitly: using the data provided by the surrounding environment of the user such as the location that can be inferred by different means like GPS or time that can be obtained implicitly from the system itself (Palmisano et al. 2008, Chen 2004,

[3]http://ontologydesignpatterns.org/wiki/Main_Page

Gu et al. 2005, Hong et al. 2009). For instance, Karantonis et al. (Karantonis et al. 2006a) detect the user's movement using the accelerometer sensor. The work presented in (Berchtold and Beigl 2009) used the microphone and the accelerometer sensors in order to recognize the user's knock on the table as appreciation. The user behaviour can also be perceived by implicit parameters (Kelly and Fu 2007) such as clicks history, browsing data or eyes movement (Shen et al. 2005, Teevan et al. 2005).

- Inferring: contextual information can be inferred using data mining and statistical methods applied, for instance, on the user's browsing history (Eirinaki and Vazirgiannis 2003, Mobasher 2007). For example, we can apply data mining approaches such as classification on a set of watched TV programs and the visited channel related to a given user in order to figure out the user's preferences (Webb et al. 2001).

2.5 Context dimensions

Dimensions of context for context-aware applications have been widely addressed. Many works used location as an approximation of context. However relying exclusively on location cannot explain the entire context of a user. Schilit et al. (1994) showed that context should make use of the changing aspects of the environment in terms of the user surroundings (light, noise, ...) and the computing environment. Nevertheless, according to Schmidt et al. (1999a), the computer environment overlaps with the physical environment. Thus, they proposed a hierarchical context model where context is divided into general categories of human environment and physical environment. These general categories are also divided into three sub-categories where a set of relevant features is identified. The value of these features will determine the overall user's context. Based on Schilit et al. (1994) and Schmidt et al. (1999a) works, researchers began to propose their own dimensions of context by expanding, reducing the dimensions or proposing new general models

of context according to their own works. Tamine-Lechani et al. (2010) summarized in figure 2.1 the different context dimensions combination for the Information Retrieval task.

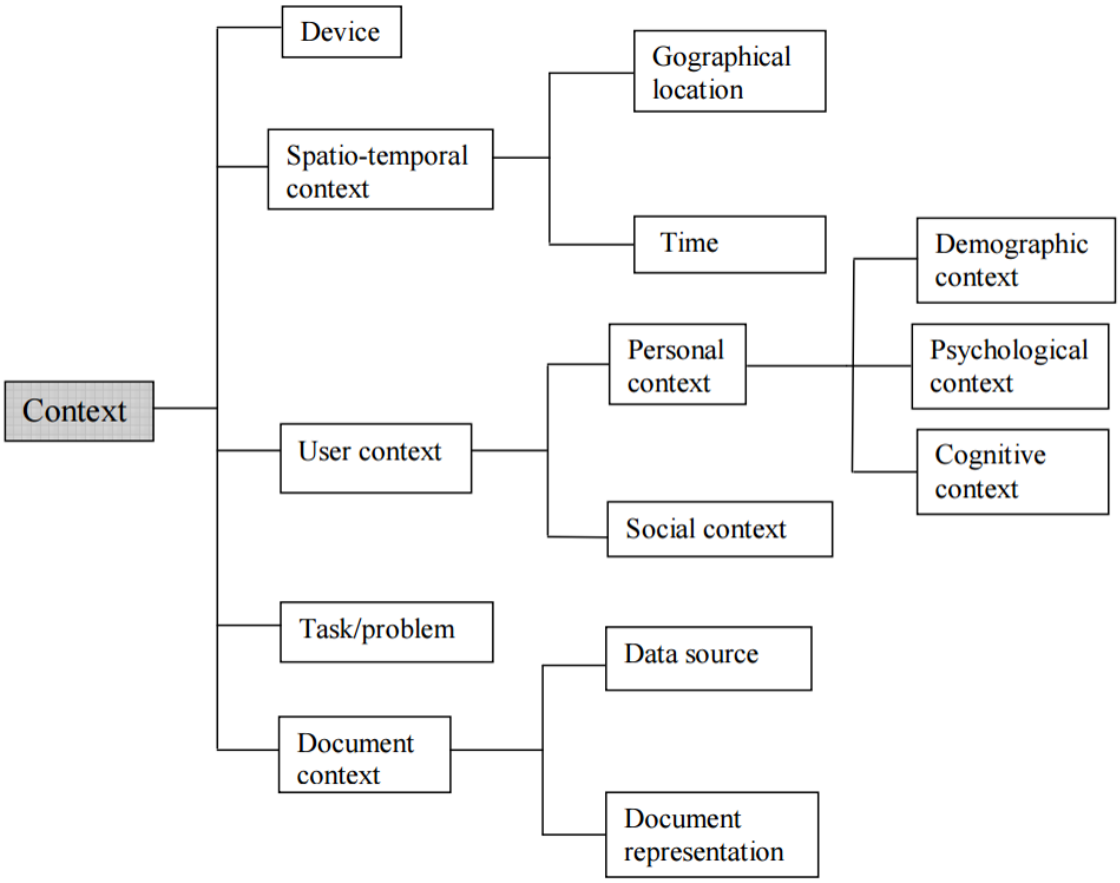


Figure 2.1: The multi-dimensional concept of context in IR (Tamine-Lechani et al. 2010)

Alidin and Crestani (2013) proposed their own dimensions of context inspired from different approaches and they merged them with the embedded sensors in Apple iPhone. They take into account the following dimensions:

- User's profile - this dimension stores any information about the user. Information such as who the user is and habits are saved in this dimension.
- Time - this dimension contains the date, the day of the week and the time of the day.
- Location - this dimension stores the information about where the user is.

- Sounds - this dimension identifies the surrounding sounds where the user currently is.
- Activity - this dimension identifies and stores user's activity.
- Agenda - this dimension contains user's driven data on user's future activities.
- Speed - this dimension indicates any change in speed if the user is on the move.
- Heading - this dimension updates user's heading in order to recognize if the user has the possibility of visiting previous location in his context.
- Network - this dimension indicates if the user is connected to the Internet.
- Preferences - this dimension refers to user's interest in some particular topics.

These dimensions are continuously interpreted in order to capture the user context. The interpretation is based on a model organized into the following context levels:

- User's scenario: situations encountered by a particular user.
- High-level context: a description of user's current context. User's current context is characterized by interpreting multiple context dimensions.
- Context dimensions or low-level context: a characterization of multiple sensors data into meaningful information. It is recognized that one dimension of a context is a subset of high-level context.
- Sensors data: any information collected from embedded sensors in the smartphone and information from user's interaction with the mobile applications.

Coppola et al. (2005) modelled context for their system, consisting in pushing mobile applications to the mobile devices (cellular phones, smartphones, PDAs, etc.), on the basis of the current context the user is in, with data received through:

- Physical sensors: Almost all mobile devices are equipped with some form of wireless network technologies (GSM, GPRS, Edge, UMTS, Bluetooth, Wi-Fi, Radio Frequency, IrDA, etc.), and can therefore sense if there is a network connection around them (and the strength of the corresponding electromagnetic field). Moreover, the device might be equipped with sensors capable of sensing data about the physical world surrounding the mobile device (e.g., noise, light level, temperature, etc.), some of which might be sent to the device by surrounding sensors.

- "Virtual" sensors: They are other processes running on user's mobile device, like an agenda, a timer, an alarm clock, and so on.

- Context sensors: They are context information provided by a server that pushes information about the current context to the users' device, with the aim of providing a more precise and complete context description. This server might be implemented by a WiFi antenna, an RFID tag sensed by the mobile device, or any other technology.

- Explicit user actions: The user can explicitly communicate, via the user interface, data about the current context. For instance, he/she might choose a connection/network provider; set the alarm clock, select the silent mode, and so on.

As we are working within a mobile context, we describe in the following sections the most commonly representing factors of the contextual information.

2.5.1 The user's profile

The user profile is an important dimension considered within contextual information. Indeed, it aims at representing and evolving the user information needs in the short and medium terms. This issue is, in itself, a double challenge consisting in translating the user interests on one hand and bringing out their diversity on the other hand.
The user's profile can be explicitly expressed or learned implicitly using for instance the

user's browsing tendency or his/her social interactions (Shmueli-Scheuer et al. 2010). For example, Li et al. (2012a) took advantage of the functionalities offered by twitter and foursquare[4] to support the users' points of interest. Indeed these two social networks enabled their users to tag their tweets with more accurate high-level geo-information. The authors explored the place level geo-information arising in twitter and foursquare in order to predict users' likelihood of visiting a place based on their current and previous visits. They believe that, with this kind of knowledge, advertisers could display targeted information more accurately to earn more attention and clicks.
For example, having known a user would go for a coffee after work, a suggestion of an attractive lately-opened cafe would be more persuasive before he/she already visited one. The user profile defining process can be characterized by three phases (Tamine et al. 2007). The first phase focuses on the representation of information units representing the profile. The second phase is related to the instantiation of the model during a particular activity. Finally, the third phase concerns the profile changing over time. Each of these phases involves approaches and techniques of representation and/or construction and are summarized below.

2.5.1.1 The user's profile acquisition

The basic model most commonly used for the representation of the user interests is the vector space model in which each interest is represented by a representative list of terms. However, there are three main representation approaches: set-based, semantic and multidimensional.

- **Set-based representation**:
  The profile is generally formalized as vectors of weighted terms (Budzik and Hammond 2000, Dumais et al. 2004) or as non-hierarchical vector classes (Gowan 2003, Pazzani et al. 1996) or as hierarchical (Kim and Chan 2003) in order to take into account multiple points of interests. The non-hierarchical representation considers

[4]https://foursquare.com

the interests as independent in the profile description and their possible dependence can be taken into account during the integration of the profile in the documents relevance evaluation phase. On the other hand, the classes' hierarchy representation can translate specificity/generality relationships between interests.

- **Semantic representation**:

  The profile representation, in this case, highlights the semantic relations between information. Representation is essentially based on the use of ontologies (Gauch et al. 2003, Challam 2004, Nanas et al. 2003, Liu et al. 2004) or probabilistic semantic networks (Lin et al. 2005, Wen et al. 2004).

  In this approach, the user interests are matched to the ontology concepts domains. A profile is then represented by ontology concepts relevant to the user. The reference ontologies used in this context are based on the general hierarchy categorization of Yahoo, Magellan, Lycos and ODP (Open Directory Project).

- **The multidimensional representation**:

  It is structured according to a set of dimensions, represented by various formalisms (Amato and Straccia 1999, Kelly 2004). The P3P[5] standards proposals for securing profiles have defined classes distinguishing user demographic attributes (identity, personal data), the professional attributes (employer, address, type) and the behavioural attributes (history of navigation).

  In this context, Amato and Straccia (1999) expose a profile representation model structured in predefined dimensions (or categories): the personal data category, the gathering data category, the delivering data category, the actions data category, and the security data category. The author proposed this model in the framework of the development of an advanced Digital Library service.

[5]https://www.w3.org/P3P/

2.5.1.2 The user's profile construction

The profile construction reflects a process that allows to instantiate the representation from various information resources. This process is generally implicit and based on the context and the user preferences inference process through the user behaviour when using:

- An IR system (Kelly 2004, Gauch et al. 2003): queries and documents explicitly or implicitly deemed relevant (consulted and/or printed and/or saved etc.).
- A web browser (Gowan 2003, Armstrong et al. 1995): explored links, recently visited pages, etc.
- Other applications (Budzik and Hammond 2000, Gowan 2003, Dumais et al. 2004): the desktop application, e-mail tools, etc. The information extracted from these resources is organized according to the profile representation model using different techniques. The most common one is based on the text statistical analysis according to the Rocchio algorithm (Rocchio 1971).

  Another technique widely used in the profile construction process is the classification, applied on the information collected from the user.

However, most of these systems require that users express their interests and input keywords or tags which is, most of the time, inconvenient in a mobile environment since it entails extra efforts from the user such as tagging, searching, or clicking (Nguyen and Riedl 2013). Mobile systems can help keep track of user's activities, preference and location.

Besides, various systems relied on the user's past or actual behaviour history to determine the user interests. Behaviour stands for Web browsing history/clicks (Shmueli-Scheuer et al. 2010, Das et al. 2007); previous visiting behaviours for location based systems (Li et al. 2012a, Pu et al. 2012) and previous reading patters for news recommendation systems (Lee and Park 2007a, IJntema et al. 2010, Gershman et al. 2011, Arora and Shah 2011, Athalye 2013, Dumitrescu and Santini 2012). However, we cannot only rely on the user's past behaviour since it may not contain enough information about the user's

interests. A user can browse a web page by chance or visit a particular place without having a real interest in that place. A user can simply open a document to work on without being related to it in any way or have a conversation about an issue that he/she is not concerned to know any information about. Furthermore, the social activity of a user provided through comments posted in his/her social accounts wouldn't possibly provide enough information to build a balanced profile since it might be limited or even missing. This kind of user profiling contains a lot of noise to manage and not enough information to build a balanced user profile.

2.5.1.3 The user's profile evolution

The profiles evolution means adapting the profile to changes in the user points of interests, and therefore in his information needs. The evolution phase only makes sense when the profile has a permanent structure, which allows distinguishing the short-term needs, constructed from the current interaction session, and the long-term needs that are a real representation of the user persistent points of interests.

Tamine-Lechani et al. (2010) noticed that little work has explored the problem of the evolution of the user profile in terms of the temporal dimension (short term, long term). They added that the profile evolution is approached as a representation problem of the diversity of the user points of interest using clustering techniques (Pazzani et al. 1996, Mizzaro and Tasso 2002, Gowan 2003) or heuristics related to the concept of the artificial life cycle of a point of interest (Chen and Sycara 1998). Kacem et al. (2014) integrated the concept of "*freshness*" in order to adjust the weights assigned to the user's profile terms with a temporal feature. Their approach helps to capture the long-term and the short-term user's profile.

2.5.2 Location

The Global Positioning System (GPS) integrated or installed in the device helps to define the user's location. This location is displayed, according to latitude and longitude. Those

GPS coordinates are not the only features that we can consider when defining a location. Indeed, as discussed by Dobson (2005), there are different ways to characterize the location of the mobile user :

- Absolute position
- Relative (next to, ...)
- A Place name
- A named class that represents the type of the place, eg. museum, school, ...

The place type can be recovered using a GIS (Geographical Information System) such as geonames[6] or foursquare which assign a location category (restaurant, train station, etc.) to a given GPS coordinates. The localization accuracy helps to determine the user's context in a more precise manner (Christoph et al. 2010).

2.5.3 Time

In the Merriam-Webster dictionary[7] , time is defined as "*a non-spatial continuum that is measured in terms of events that succeed one another from past through present to future*". Time can also be expressed as "*the measured or measurable period during which an action, process, or condition exists or continues*".
According to the second definition, several time units have been defined, e.g. hours, days, months and years (Whitrow 1989). The time conception and measurement flexibility implies different representations of time context information.
Indeed, time may be represented as a continuous variable whose values determines the specific times at which items are rated by a given user.
Example: `user A rated item I at t = June 1st, 2010 at 18:05:00`
Another way to model time is to identify categorical values, for the time periods of interest

[6] http://www.geonames.org/
[7] http://www.merriam-webster.com/dictionary/time

(Ressad-Bouidghaghen 2011). For example, in the tourism domain (Gavalas and Kenteris 2011), the variable "`season`" can be expressed as:

$season = \{hot_season, cold_season\}$.

Time can also be modelled in a hierarchical way which makes possible to define the degree of granularity of the time context information (Campos et al. 2014). Example:

$Week_{Day} = \{Monday,\ Tuesday,\ .\ .\ .\ ,\ Sunday\} \Rightarrow time = \{morning,\ afternoon,\ ...,\ night\}$

2.5.4 The user's activity

Verbert et al. (2012) consider that the user's activity reflects the objectives, the tasks or the actions of a user. The tasks or the actions, the user performs at a given time, are commonly used in literature as triggers to launch the recommendation process. Works on the task-based recommender systems generally used the user's current or past behaviour history. Zhang et al. (2015) predict the user's information need given the user's browsing history and the current browsing session. They classify the browsing session into "*demand sequences*" related to specific categories. For example :

"*Amazon and eBay are for the same demand of online shopping*"

"*Youtube relates to online videos demand*"

Several other works made also use of the user's web search log in order to extract information about activities that a user may undertake (Jansen and Spink 2006, Richardson 2008, Orlando and Silvestri 2009, Tolomei et al. 2010). The information extracted are minded to figure out the user's past and current activities. Chen et al. (2000) described the user's activity through three different schemes:

- Machine vision: using image processing and camera technology (Kern et al. 2003a, Bao and Intille 2004, Mathie et al. 2003, Nakata 2006). The work presented by Cho et al. (2008) put forward an approach that aims at detecting the user's activity to monitor his/her medical status using a wearbale camera and an accelerometer.

- The user's calendar: to figure out what are the different activities scheduled at a certain time. Bouneffouf (2013) considered that the user's activity is inferred from the calendar integrated within the user's mobile device. They assume that the user indicates, for each activity he/she may undertake, the time and the location information, which is not always the case in real life.

- Artificial Intelligence techniques: that help to determine contextual information by leveraging low-level sensors. Schmidt et al. (1999b) used the accelerometer and the light sensors in order to detect if the user's device is "*in hand*", "*in a suitcase*" or "*on a table*". This approach aims at defining the user's behaviour regarding his/her mobile device and then infer a certain aspect of the user's activity.

Indeed, the user activity can be depicted from the different application and sensors installed in the mobile device. The data provided by the sensors can be saved in context logs in the mobile device or sent to the server.
In the following section, we describe some of the sensors that are commonly used to determine the user's activity.

Camera The camera is basically used for image recognition (Luley et al. 2005). Visual object detection from mobile phone imagery for context awareness helps to give an overview of the user's surroundings and the images that the user have been capturing.

Accelerometer An accelerometer is a sensor attached to a mobile device or any other object, used to measure the linear acceleration of the given device[8] . This sensor helps to classify movement pattern into specific situations where, for example, *the user is walking* or *sitting down* or *running* (Karantonis et al. 2006b, Kern et al. 2003b).

Microphone The microphone is essentially used to detect the volume of the noise in the user's surroundings in order to identify places according to the background noise

[8]https://en.wikipedia.org/wiki/Accelerometer

emitted.

The work of Ma et al. (2003) showed an overall accuracy of 91.5% for being able to detect whether the user is in office, at the beach or at a football match. The microphone can also be used to figure out how many persons are in a particular place (Christoph et al. 2010).

Compass The electronic compass initially integrated in mobile devices or just installed as an application helps to overcome the GPS and the accelerometer limitations and support them by giving more information about the user's movement. Indeed, the GPS only delivers the user's position but it is unable to detect the user's direction and it does not work indoors.

However, the activity recognizing approaches that are based exclusively on sensors, follow usually a predefined model and fail to detect a model that has not been predefined. Besides, users are unwilling to use wearable sensors on a daily basis (Motti and Caine 2015).

2.6 From context-aware systems to situation-aware systems

The context concept is perceived as the cognitive, the social and the professional environment related to several factors like time, locations, etc. The use of these factors is very crucial to boost the performance of any system, however, they only form a low-level layer extracted from embedded sensors that need to be interpreted into a high-level layer that defines a situation.

As expressed by Bouneffouf (2013), "*Situation awareness focuses on the modelling of a user's environment to help his/her to be aware of his/her current situation*".

The notion of situation takes part in the Situation Calculus Theory introduced by McCarthy in 1963 (McCarthy and Hayes 1969, McCarthy 1963) and was defined as "*the*

complete state of the universe at an instant of time". Reiter (2001) proposed an action theory formalism defining the situation notion and considered it as "*A finite sequence of actions. It is not a state, it is not a snapshot, it is a history.*"

Endsley (1995) perceived the situation concept from two different angles: formal and informal. The formal definition considers that a situation is "*knowing what is going on*". The informal definition regards a situation as "*the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future*".

As it comes to pervasive systems[9], several works associated the situation concept with context and its dimensions as the following :

"*The situation of an application software system is an expression on previous device-action record over a period of time and/or the variation of a set of contexts relevant to the application software on the device over a period of time. Situation is used to trigger further device actions*" (Yau et al. 2004).

"*A situation is a set of semantic relations between concepts (in one context dimension or between several context dimensions) which are valid and stable during an interval of time*" (Bouzeghoub et al. 2007).

These latter definitions seem appropriate regarding the framework we are working within since we perceive a situation as an interpretation of an instantiated set of contextual dimensions, such as time and location. Indeed, the raw data recovered by the sensors embedded within the mobile device are abstracted to determine a high-level contextual information that allow to put forward the user's situation.

In the literature, there are several works that proposed their own context dimensions combination frame in order to infer the user's situation. Ressad-Bouidghaghen (2011) used a vector space model that combines the context factors within a personalized Information Retrieval task. They consider that the user's situation is represented by time and location and particularly : *the type of location*, *the season*, *the day of the week* and *the time of the*

[9]https://en.wikipedia.org/wiki/Ubiquitous_computing

day. McCall and Trivedi (2007) proposed a situation-aware driver assistance application that takes into account the road infrastructure, climatic information and the driver related information in order to predict critical situations that may encounter the driver, like accidents, and help the driver get through them.
In the following chapter, we detail the several works undertaken within the situation-aware recommendation domain.

2.7 Conclusion

In this chapter, we have introduced the basic concepts related to context and the different aspects and dimensions that this notion entails. We presented the several works dealing with the various types of context, ranging from the user's personal context to the spatio-temporal context. We exposed the different techniques of context-acquisition, representation and exploitation within context-aware systems. As reported, we notice that each system defines its own context model according to the field of application considered. As we are working within the context-aware recommendation domain, we detail, in the next chapter, how contextual information is employed to boost the performance of recommender systems and particularly the proactive ones. We also put forward the novel concepts tackled by such systems, to mention the intrusiveness aspect.

Chapter 3

Recommender Systems

3.1 Introduction

Recommender systems have become an independent research area in the mid 1990s (Goldberg et al. 1992). With the development of web platforms and new technologies, the interest in recommender systems has significantly increased and has spread to cover multiple domains such as movies[1], tourism[2] and videos[3]. GroupLens (Resnick et al. 1994) is considered as the first automates movie recommender system followed by a music recommender system called RINGO (Upendra 1994) and BellCore (Hill et al. 1995) for video recommendation. Traditional recommender systems' aim is to provide relevant information to users. However, with the recent spread of mobile devices (smartphones and tablets), we notice that recommender systems are progressively adapting to pervasive environments in order to deliver not only relevant information to users but also when it is most needed. Indeed, the amount of the contextual information provided by the mobile devices sensors such as temperature, GPS, accelerometer, etc, help to understand the users' needs and to deliver recommendations without the user's request. This is called context-aware proactive recommendation.

[1]Netflix https://www.netflix.com/
[2]Tripadvisor https://www.tripadvisor.com/
[3]Youtube https://www.youtube.com/

In this chapter, we present the basic concepts related to recommender systems. We also detail the proactivity and the intrusiveness notions integrated within the recommendation process.

3.2 Definition

Gaillard (2014) considers that the beginning of the recommendation research area can be related to works in forecasting theory (Armstrong 2001) and Information Retrieval (Salton 1989).

The works presented by Resnick and Varian (1997) and Herlocker et al. (2000) define a recommender system as a system that is capable of learning user's preferences in order to provide new items that might be interesting to the user.

According to Burke (2002), a recommender system must be able to provide individualized recommendations and guide users in a personalized way. Burke's definition adds new notions such as individualization and personalization.

Meyer (2012) put forward a 4-key features description regarding recommendation. He assumed that recommendation is related to 4 key actions:

- "*help to decide*": be able to predict an item rating
- "*Help to compare*": present a personalized ranked list of items to a user
- "*Help to discover*": recommend unknown items that are deemed to be relevant to user's tastes
- "*Help to explore*": provide items that are similar to a particular item

Sharma and Mann (2013) present a formal definition for recommender systems expressed as the following:

$$\forall c \in C, S_c = \arg\max_{s \in S} \quad u(c, s) \tag{3.1}$$

Where:

C is the set of users

S is the set of items that can be recommended

U is a utility function that computes the usefulness of an item s regarding a given user u The item utility is generally presented by a score or a rating, measuring to what extent a user liked the given item. The rating is usually presented on a 5-point scale where 1 stands for dislike and 5 means a like. The utility can also be measured using the user's characteristics such as the age, the gender, preferences; compared to the item's characteristics like number of stars and location for a restaurant recommender system.

3.3 Recommendation approaches

Burke (2007) provided a general taxonomy of the recommender systems types that has been adopted as a reference in this research area. He assumed that there are three main approaches:

- Content-based approach: The system recommends items that are similar to those liked by the user in the past. The items' similarity is calculated using the characteristics associated with the compared items. For example, if the user has positively noted a book that belongs to the genre "thriller", the system can recommend this kind of books.

- Collaborative approach: The system recommends to the user items that other users with similar tastes liked. The users' similarity is measured using their rating history.

- Hybrid approach: The system combines the above-mentioned approaches

3.3.1 Content-based approaches

Content-based recommender systems attempt to recommend items that are similar to those liked by the user in the past. Indeed, the key idea is to match the user profile features describing the user's preferences with the items features in order to recommend new interesting items (see Figure 3.1).

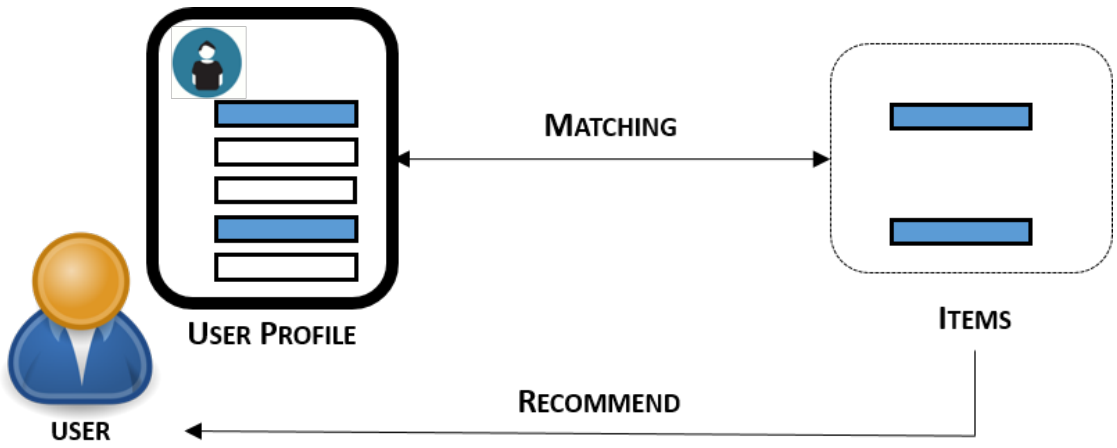


Figure 3.1: Content-based approach process

This process takes part in two main research areas which are Information Retrieval (IR) and Artificial Intelligence (Baeza-Yates et al. 1999).

The information search process in IR is quite similar to the one entailed in recommendation. The only difference resides in the fact that the user does not explicitly express his/her needs by providing a query. In recommender systems, the user's need is represented by the user profile learned from the various activities undertaken by the user. It usually involves the application of Machine Learning techniques that help to figure out if a new information is interesting or not based on the information that were considered interesting by the user.

In the following sections, we browse the different items representation techniques and the recommendation algorithms used.

3.3.1.1 Representation techniques

The items that can be recommended to users are represented by a set of characteristics, known as attributes or features that depend on the type of the item to recommend. For example, in a book recommender system, the features that might be used to describe the item (i.e. book) are: *author*, *genre*, *subject*, etc. The items' features are usually textual and are extracted from webpages, news articles or product descriptions (Picot-Clémente 2011).

Picot-Clémente (2011) considers that the construction of a user profile by analysing textual features is somehow complicated since Natural Language is ambiguous. The keywords profiles are not able to capture the users' interests semantics because they are mainly generated by a term matching process. If a term is found in both the profile and in the document, a match is made and the document is considered appropriate. This technique suffers from polysemy (multiple meanings for a word) and synonymy (different words with the same meaning) problems that induce the recommendation of non-relevant documents. Thus, semantic analysis techniques are deemed more appropriate to tackle these problems.

3.3.1.2 Recommendation algorithms

3.3.1.2.a Keywords-based recommendation

Most of the recommender systems use the Vector Space Model with the basic TF-IDF weight (Term Frequency-Inverse Document Frequency). This model is a semantic aware document representation introduced by Salton et al. (1975). A document is represented by a weighted vector of terms where weights indicate the degree of association between document and its terms. For the following, we denote:

The corpus documents: $D = d_1, d_2, ..., d_n$

The corpus set of terms : $T = t_1, t_2, ..., t_m$

T is constructed by applying simple techniques like tokenization and stemming (Baeza-Yates et al. 1999).

Each document d_i is represented by a vector in a vector space of m dimensions:
$d_i = w_{1i}, w_{2i}, ..., w_{ji}$ where w_{ji} is the weight of the term t_j in document d_i
The term weighting scheme most commonly used is the TF-IDF (Term Frequency-Inverse Document Frequency) based on assumptions (Singhal and Salton 1995) that:

- "*a term that occurs frequently in a text is more important in the text than an infrequent term*" - TF
- "*the more documents a term occurs in, the less important it may be*" - IDF
- "*the term frequency factors may be large for long documents... unfairly, increasing the chances of retrieval over shorter documents ... Normalization is a way of imposing some penalty on the term weights for longer documents*" - Normalization

In other words, the words that appear frequently in a document but rarely in the rest of the body are more likely to represent the subject of the document. Besides, normalization compensates the fact that long documents have more chance to be found.

$$TFIDF(t_k, d_j) = TF(t_k, d_j).log\frac{n}{n_k} \tag{3.2}$$

Where n is the number of documents in the corpus and n_k is the number of documents in the corpus in which the term appears at least once.

$$TF(t_k, d_j) = \frac{f_{k,j}}{max_z f_{z,j}} \tag{3.3}$$

The maximum is calculated based on the frequency $f_{z,j}$ of all the terms t_z that appear in the document d_j. Then, the term weight is obtained using the cosine normalization that helps to define a weight in a [0,1] interval:

$$W_{k,j} = \frac{TFIDF(t_k, d_j)}{\sqrt{\sum_{k=1}^{|T|} TFIDF(t_k, dj)^2}} \tag{3.4}$$

Then, in order to measure the documents relatedness, a similarity measure must be defined. The most widely used one is the cosine similarity:

$$sim(d_i, d_j) = \frac{\sum_k w_{ki}.w_{kj}}{\sqrt{\sum_k w_{ki}^2}.\sqrt{\sum_k w_{kj}^2}} \tag{3.5}$$

For content-based recommender systems using the vector space model, user profiles and items are represented as weighted vectors. The utility of an item to a given user may be calculated by the cosine similarity between user profile vector and item vector.
Key-words based recommender systems were developed in various fields of applications, such as news, restaurants, movies, etc. Personal WebWatcher (Mladenic 1999) learns the user's interests using the web pages that the user visits. The same approach is adopted by Moukas (1997) where specific filtering agents are used to assist users in finding information. The user can define the filtering agents by providing web pages (represented as weighted vectors) that are closely related to his/her interests.
YourNews (Ahn et al. 2007) is a news recommender system that considers a separate interest profile for 8 different topics (Business, World, National, etc.). The user's interest profile for each subject is represented by a vector of weighted terms extracted from snippets of news already seen by the user. The latest news articles viewed by the user are collected, and the 100 most-weighted terms are extracted to generate the final vectors. The system considers the short-term profile, including only the last 20 articles, while the long-term profile use anything that has been viewed.

3.3.1.2.b Semantic-based recommendation

Semantic based recommender systems draw on methods that make use of the semantic of the items being handled. SiteIF (Magnini and Strapparava 2001) was the first system to make use of the meaning of documents to build the user's interests model. The system uses MultiWordNet[4], a multilingual lexical database to associate a list of sets of synonyms

[4]http://multiwordnet.fbk.eu/english/home.php

(called synsets) with news articles. The user profile is a semantic network where nodes represent the synsets extracted from the news articles read by the user. A matching phase is then conducted in which the system receives as input the document synsets and the user model, and outputs an estimation of the document relevance (Stefani and Strappavara 1998). ITR (iTerm Recommender) (Degemmis et al. 2007, Semeraro et al. 2009) provides recommendations for several types of items (music, movies, books), assuming that the item's descriptions are available as text documents and uses the WordNet lexical ontology[5].

Quickstep (Middleton et al. 2004) is a recommender system for research articles. The system makes use of an ontology based on the DMOZ open directory project[6] (DMOZ open directory project) scientific classification (27 classes were used). The key idea is to associate the articles with an ontology class using the KNN classifier. Thus, the user's profile contains a set of topics and the interest degree related to them. The item-profile matching is conducted by calculating the correlation between the first three interesting topics in the user's profile and the papers that belong to these topics.

News@hand (Cantador et al. 2008) is a news recommender system that makes use of an ontology of item characteristics and user's preferences. The annotation process associates news articles with concepts belonging to domain ontologies such as politics, education, sports, etc. The items' descriptions are represented by TF-IDF scores vectors based on the set of concepts defined in the ontologies. The user's profile is similarly represented. The item-profile matching is determined by the cosine similarity.

3.3.2 Collaborative-based approach

Unlike the content-based filtering approach, that uses the items previously rated by just one user, the collaborative filtering approaches consider multiple users' ratings. The key idea is that the score that the user may give to a new item is likely to be similar to the one given by another user, if these two users have previously scored other items in a similar

[5]https://wordnet.princeton.edu/
[6]http://www.dmoz.org/

manner.

Collaborative filtering approaches overcome certain limitations encountered by content filtering approaches in a way that the items which content is not defined, or difficult to define may still be recommended to users through the feedback of other users. Moreover, unlike content-based systems, collaborative filtering can recommend items with different contents, as long as other users liked these items (Picot-Clémente 2011).

Collaborative methods can be gathered into two general classes: neighbourhood-based methods and machine learning-based methods (Cantador et al. 2008).

3.3.2.1 Neighbourhood-based recommendation

This recommendation approach automates the word-of-mouth principle that considers the opinions of users who share the same interests (Picot-Clémente 2011). Thus, the previous users' ratings are used to predict ratings for new items using one of these two ways: user-user recommendation or item-item recommendation.

- *user-user recommendation* :

  This approach assesses the interest of a user u for an item using the rates given by other users, called neighbours, regarding this item. These users (i.e. neighbours) have similar rating patterns like user u (Konstan et al. 1997, Hill et al. 1995, Shardanand and Maes 1995). The user's rating on an item can be predicted by the average rates of the user's neighbours.

$$\hat{r}_{ui} = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{vi} \tag{3.6}$$

  where:

  $N_i(u)$ is the set of the k-nearest neighbours of user u that have rated item i

  r_{vi} is the rate of item i given by user v

- *Item-item recommendation* :

  This approach predicts the score of a user u for an item i using the rates given

by u to items similar to i (Linden et al. 2003, Deshpande and Karypis 2004). The predicted score of user u on item i can be obtained by measuring the average of the scores given by u to similar items to i:

$$\hat{r}_{ui} = \frac{1}{|N_u(i)|} \sum_{j \in N_u(i)} r_{uj} \tag{3.7}$$

where:

$N_u(i)$ is the set of items that are similar to item i and that have been rated by user u

r_{uj} is the rate of item j given by user u

3.3.2.2 Machine learning based recommendation

Unlike the neighbourhood-based systems that use previous rates for prediction, model-based approaches consider the users' rates to build a learning model. It is about modelling the user-item relation with representative features.

These systems were developed using machine learning techniques such as bayesian clustering (Breese et al. 1998), support vector machine (Grčar et al. 2006), and the singular value decomposition (Bell et al. 2007a, Koren 2008, Paterek 2007). These systems describe both items and users by features to compare them directly. It is about measuring to what extent a user might like an item on each particular feature. Recommendation can also be tackled as a classification problem to figure out what might interest the user and what might not. For this task, a variety of algorithms are used, such as decision trees, clustering, neural networks and bayesian classifiers. Daily Learner (Billsus and Pazzani 2000), adopts two separate user models. The first is based on the nearest neighbour text classification algorithm to learn the short-term user's interests, while the second, is based on a naive bayesian classifier using data collected over a long period of time to represent the user's long-term interests. Mooney and Roy (2000) implement a naive bayesian based method for recommending books exploiting the description of the products obtained from

Amazon[7] .

This kind of approach offers high expressive abilities to describe the various aspects of the data. Thus, it tends to provide more accurate results than neighbourhood-based systems. However, usual commercial systems like Netflix[8] and Amazon[9] prefer to use the neighbourhood-based approach because they consider it easier and intuitive to handle (Picot-Clémente 2011).

3.3.3 Hybrid approaches

It is known that the major problem encountered by content and collaborative based methods is the cold start problem (Das et al. 2007, Chu and Park 2009, Adomavicius and Tuzhilin 2005) that makes difficult for recommender systems to tackle users' preferences without a prior knowledge about their rating patterns.

Thus, several systems tried to combine these two approaches to figure out a solution to this problem. Indeed, in the Netflix competition, the winning candidate adopted a hybrid approach (Bell et al. 2007b).

Burke (2007) classified hybrid recommender systems into 7 groups according to their combination techniques:

- Weighted recommenders: combine different recommender systems' scores
- Switching recommenders: the recommender system switches from a recommendation approach to another depending on the best recommendation strategy
- Mixed recommenders: they are similar to the weighted recommenders scheme but they differ on the way they present recommendation to the user in different lists
- Feature-combining recommenders: they are considered as a meta-recommender system having as input, information from many recommenders

[7] https://www.amazon.com/
[8] https://www.netflix.com/
[9] https://www.amazon.com/

- Cascading recommenders: the input of a recommendation algorithm is the output of another one
- Feature-augmenting recommenders: the recommendation algorithm's results are considered as a feature input to another recommender.
- Meta-level recommenders: they use one algorithm to train a model, then proceed with using the whole model as input for another recommendation algorithm.

There are also different hybridization combination strategies that were developed:

- Implement content-based and collaborative-based approaches separately and then proceed with their predictions combination
  Claypool et al. (1999) developed a news recommender system that uses separately the two approaches but combines them lately in an average weighting scheme that figures out the optimum combination for each user.
- Run alternatively either CBF or CF first
  Joachims et al. (1997) developed a recommender system that uses the user's log file as well as the past behaviour of other users to recommend web pages when a user is browsing. Cotter and Smyth (2000) proposed a TV recommender system that represents the user's profile using the watched TV shows and then generate a list of items using both content and collaborative methods.
  Rojsattarat and Soonthornphisaj (2003) also measure users' profiles similarity using all the items that have been assessed and not only the documents in common. This method tackles the possibility of not having common items assessed by users. Sarwar et al. (2000) applied singular value decomposition as a dimensionality reduction technique to content-based profiles in order to compute the likeliness of products by customers.
- Construct a unified model that incorporates the two techniques
  A general unified model described by Zhang and Koren (2007) learns users' profiles

using their feedback and information borrowed from other users through a bayesian hierarchical model that classifies items as interesting or non-interesting.

A mobile recommender system for blog articles was presented by Liu et al. (2011) combining 3 different techniques: topic clustering using TF-IDF to gather articles into clusters; attention degree predicting the topic clusters popularity; and collaborative filtering predicting the user's ratings using previous history.

3.3.4 Recommender systems issues

The recommender systems area has considerably progressed throughout the years along with the technological advance. However, there are still many issues and challenges that need to be addressed and are actually tackled as a research topic.

The major limitations challenging recommender systems are reviewed as the following.

- **Cold-start problem**

  The cold start problem happens when the system deals with a user whose preferences and ratings are missing or when there is no information about an item. Most of the collaborative recommender systems encounter this issue as they essentially use information about the user and the item. Content based approaches are less affected and just face this problem at the user level when this latter has no previous ratings or behaviour.

  Rashid et al. (2002) tried to deal with the cold start problem at a user level and within a collaborative filtering approach by suggesting to users the most valuable items in terms of information value so they can be rated by the users and thus infer the users' preferences.

  The work presented by Massa and Bhattacharjee (2004) makes use of the "*user's web of trust*" (i.e. the users that the actual user trust) in order to recommend items to cold-start user. Park and Chu (2009) developed a predictive feature-based regression approach that run through all the information that can be possibly gathered about users and items like demographic information.

As pointed by Chamsi Abu Quba (2015), the recommender industry tries usually to solve the cold start problem by suggesting to new users the most popular items and wait for the user's action regarding these items to build the user's profile.

- **Sparsity problem**

  The recommendation quality is very influenced by the sparsity problem and especially the collaborative-based approaches as they usually use a user-item ratings' matrix that becomes sparse as the number of items and users increases. The major issue is that users do not rate all the items and when they do, the ratings are usually sparse (Sharma and Mann 2013).

  There are several works (Zhou and Luo 2010, Desrosiers and Karypis 2008) that were proposed to alleviate this problem, to mention the one presented by Zhou and Luo (2010), in which the authors use Multiple Imputation technique to fill in the missing rating values with plausible values representing uncertainty. Despite the several approaches that were proposed to deal with the sparsity problem, there is still a demand for more research.

- **Over-specialization** Recommender systems and particularly the content-based ones encounter the over-specialization issue as they tend to recommend items that are very similar to those already known by the user (Adamopoulos and Tuzhilin 2014, Chen et al. 2011). This problem is known as "*The Harry Potter effect*" (Koolen et al. 2015). Since a lot of people seem to like the Harry Potter movie, this latter scores high similarity with the other watched movies and thus hides movies that might be more interesting to recommend. As a solution, Bickson (2012) proposed to normalize the number of ratings in order to provide users with a diverse choice of items.

- **Domain dependency** Many of the actual contextualized systems are domain dependent (tourism, movies, news ...) and have specific context dimensions to apply according to the domain. However, most of them rely almost on the same context

combination which includes location ,time and user preferences with a slight difference on how to approach these dimensions. Therefore, many different services related to different domains can take advantage from the same context information without encumbering the user's mobile.

3.4 Recommender systems evaluation

Recommender systems' evaluation is performed through various methods and metrics and often depends on the type of the recommended item and the dataset used within the system.

Sandoval (2015) classifies evaluation methodologies into 3 general groups:

- Offline or online evaluation

  Usually, evaluations are conducted in offline scenarios where approaches are trained on some of the data and then compared to real data using precision and recall. It can also be conducted as users interact directly with the recommender system in real time.

- User/business oriented evaluation

  The user satisfaction is quantified according to the system requirements

- Accuracy and quality measurements

  There are other quality dimensions to consider besides the item relevance to the user, that can define the performance of a recommender system.

The following sections give more details about these evaluations methodologies

3.4.1 Offline, online evaluation and user studies

3.4.1.1 Offline evaluation

Offline evaluation is commonly used in the literature to assess recommendation approaches. This kind of evaluation depends generally on the domain of recommendation and on the

user's profile information. Several evaluation datasets were provided in different domains to mention MovieLens[10] for movie recommendation. Various works have also presented datasets for songs recommendation like the Last.fm3 (Celma Herrada 2009) and the Million Song (McFee et al. 2012) datasets.

The offline evaluation process consists in partitioning the available dataset into a training set used to infer knowledge about the user and a testing set that determines the recommendations relevance.

Sandoval (2015) proposed to classify partitioning approaches into two main classes:

- random splitting (Goldberg et al. 2001, Sarwar et al. 2001) in which the two sets are selected randomly for training or testing.
- time-based splitting (Campos et al. 2014, Gunawardana and Shani 2009) consists in picking the recent user interactions within the dataset for testing and selecting the older information for training.

After choosing the partitioning approach to adopt, there are two ways of conducting the test phase whether by predicting the rating that a user would give to a recommended item in a 5-point scale for example or by ranking a set of items (Steck 2013).

3.4.1.2 User studies

User studies (Ziegler et al. 2005, Ekstrand et al. 2014) are good alternatives for evaluating recommender systems in which users are asked to evaluate recommendations proposed by the system. This kind of evaluation allows a qualitative assessment of the system as surveys can be conducted along with the experiments. Nevertheless, user studies are not easily applicable since they are expensive and need a large users base that needs to be organized in a way that reflects different users' types in terms of age, gender, education, etc.

There are different tools that were developed to support recommender systems evaluation

[10] http://grouplens.org/datasets/

using user studies, to mention the Amazon Mechanical Turk (AMT)[11], CrowdFlower[12], ClickWorker[13]. These tools are web platforms of crowdsourcing or microworking collaborating with real persons who are paid in return for their participation in case studies (Schaffer et al. 2015, Buhrmester et al. 2011, Lee and Hosanagar 2016, Erdt 2014, Erdt et al. 2013).

3.4.1.3 Online experiments

These experiments are generally conducted with real users in real settings where a platform can redirect a part of its traffic towards the system that is being evaluated (Sandoval 2015). The system's performance is determined through metrics like click-through rate, page views (Garcin et al. 2014) and economic benefit (Shani et al. 2005).

3.4.2 User/Business-oriented Evaluation

This evaluation approach sets for a balanced performance towards users' and businesses satisfactions. Indeed, the recommender system has to make sure that users are given relevant items and at the same time increase the businesses revenue (Azaria et al. 2013). For example, Netflix avoids to recommend new released items as they are more expensive even though they might be liked by users (Shih et al. 2009).

3.4.3 Alternative quality measures

According to Herlocker et al. (2004), the accuracy metric is not sufficient to evaluate a recommender system's performance. Therefore, various other evaluation metrics were proposed such as coverage, novelty and diversity.
Chen et al. (2013), have shown that the user's personality should be taken into account in the recommendation process as it can quantify the diversity need. Winoto and Tang

[11] https://www.mturk.com/mturk/welcome
[12] https://www.crowdflower.com/
[13] https://www.clickworker.com/

(2010) have also proven that mood impacts the way users deal with recommendations. Herlocker et al. (2000) have also demonstrated that explaining a recommendation to users is very important to consider.
Diversity was also addressed as an evaluation metric as it helps to assess whether the recommendation approach is able to cover the various tastes and needs of users or not (Bollen et al. 2010, Pu et al. 2011).

3.5 Mobile, proactive and context-aware recommender systems

The overwhelming advances in mobile technologies allows recommender systems to be highly contextualized and able to deliver recommendation without an explicit request or interaction from the user at the just the right time. Just-in-time or proactive recommendation has become the new standard.
Proactive Recommendation Systems (PRSs) as described by Melguizo et al. (2007), retrieve large quantities of documents, decide what available information is most likely relevant to the user needs, and offer that information without user requests.
Ricci (2010) considers that proactive recommender systems "*can revolutionize the role of RSs from topic oriented information seeking and decision making tools to information discovery and entertaining companions*".
Several systems have been developed to support proactive recommendation and can be portioned within the following typology:

- Spatio-Temporal based systems
- The user's current or past behaviour based systems
- Activity-centric systems

3.5.1 Spatio-Temporal based systems

There are several approaches that used location as an approximation of context. The Global Positioning System (GPS) integrated or installed in the device helps to define the user's location. This location is displayed, according to latitude and longitude. Those GPS coordinates are not the only features that we can consider when defining a location. The place type can also be recovered using a GIS (Geographical Information System) such as geonames[14] or foursquare[15] which assigns a location category (restaurant, train station, etc.) to a given GPS coordinates. The localization accuracy helps to determine the user's context in a more precise manner (Christoph et al. 2010). Time was also used as a context dimension that helps to boost the recommendation relevance. It may be represented as a continuous variable, whose values determine the specific times at which items are rated by a given user, or by categorical values, for the time periods of interest. For example, in the tourism domain, the variable "`season`" can be expressed as: *hot-season* or *cold-season*. Time can also be modelled in a hierarchical way which makes possible to define the degree of granularity of the time context information. Example: $WeekDay = \{Monday, Tuesday, \ldots, Sunday\}$

The recommender systems that rely mainly on the spatio-temporal factors focus generally on a specific domain like tourism or restaurants recommendation.

Oppermann and Specht (2000) developed a system called HIPPIE that proactively recommends to users upcoming events and exhibits within a tourist user guide using indoor positioning technologies and maps. Braunhofer et al. (2015) proposed a proactive recommender system for points of interests (POI) employing mainly time and the user's visiting history of POI. The latter factor was also used by Li et al. (2012b) within a Markov chain model to predict the user's next visits. Vico et al. (2011) made use of other contextual factors like the social dimension (user alone or accompanied), besides the temporal and the geographical aspects, to proactively recommend restaurants to a user.

[14] http://www.geonames.org/
[15] https://www.foursquare.com/

3.5.2 The user's current or past behaviour based systems

To recommend items related to user's interests, various approaches depend on the user's past or actual behavior history that includes for example previous visiting behaviors for location based systems (Li et al. 2012b, Pu et al. 2012); Web browsing history/clicks (Melguizo et al. 2007) and previous reading patters for news recommender systems (IJntema et al. 2010, Arora and Shah 2011, Athalye 2013, Dumitrescu and Santini 2012).

Sae-Ueng et al. (2008) analysed the user's behavior log for shopping assistance using a digital camera and RFID sensors[16]. The system recommended information about a product according to the user's behavior classified under five states: *Standing, Viewing, Touching, Carrying,* and *Fitting.* Elbery et al. (2016) developed a carpooling recommender system that makes use of the user's past visiting history and information collected from the user's social networks accounts. The system integrated the latter aspects within a time markov chain. Quercia et al. (2010) proposed a system that automatically recommend new friends relying on colocation records to elicit relevant encounters and to arrange them into a weighted social network for recommending friends. They have engineered an application for mobile phones that silently keeps track of people's colocation, as well as frequency of voice calls and text messages. It also helps existing members to elicit new social relations, as they develop over time.

Lee and Park (2007b) presented a mobile web news recommendation system (MONERS) that incorporates news article attributes and user preferences with regard to categories and news articles. They estimate user preference of news articles by aggregating news article importance and recency (calculated by the difference between the time it was posted and the present), and a user segment that is focused on user profiles, reading patterns of news articles, changes in user interest, and usage patterns.

In the work presented by Ayachi et al. (2016), e-government services also took advantage of the user's behaviour pattern through social media, service and feedback databases to provide e-government services recommendations tailored to each citizen.

[16]https://en.wikipedia.org/wiki/Radiofrequency_identification

3.5.3 Activity-centric systems

Other approaches considered recommendation from an activity centric angle. They relied on triggers to launch the recommendation process. The triggers might take the form of ongoing conversation or activity such as text messages, phone calls (Popescu-Belis et al. 2011); opened web pages or documents (Prekop and Burnett 2003, Dumais et al. 2004, Karkali et al. 2013) and the social media activity of the user such as the content of the user's tweet stream on Twitter (Phelan et al. 2011, De Francisci Morales et al. 2012, O'Banion et al. 2012).

De Francisci Morales et al. (2012) made use of the social networks potential by developing a new methodology for recommending interesting news to users by exploiting the information in their twitter persona. They model relevance between users and news articles using a mix of signals drawn from the news stream and from twitter. This latter is used to build the profile of the social neighbourhood of the users, the content of their own tweet stream, and topic popularity in the news and in the whole twitter-land. They showed that the combination of various signals from real-time web and microblogging platforms can be a useful resource to understand user behaviour.

Phelan et al. (2011) presented a news recommendation system named Buzzer, which is capable of adapting to the conversations that are taking place on Twitter. The system uses a content-based approach to rank RSS news stories by mining trending terms from both the public Twitter timeline and from the timeline of tweets generated by a user's own social graph (friends and followers). The system also looks for co-occurrences of content between the terms that are present in tweets and RSS articles and ranks articles accordingly. Therefore articles with content that appear to match the content of recent Twitter chatter (whether public or user related) will receive high scores during recommendation.

3.6 Non-intrusive Recommendation

3.6.1 Definition

With the recent advance of mobile technologies and the rapid growth of mobile applications, the user is increasingly confronted with a lot of information. Nowadays, it is not enough for a recommender system to determine what to recommend according to user needs, but it also has to tackle the intrusiveness aspect. It is important to include the risk of disturbing the user within the recommendation process.
The Cambridge Dictionary[17] defines intrusiveness as an act:

> *"Affecting someone in a way that annoys them and makes them feel uncomfortable."*

Intrusiveness was also defined by Li et al. (2002) as :

> *"A perception of psychological consequence that occurs when an audience's cognitive processes are interrupted."*

As it comes to the recommender system domain, intrusiveness is considered by Bouneffouf et al. (2012) as a risk of disturbing the user and was presented as :

> *"The possibility to disturb or to upset the user which leads to a bad answer of the user."*

The following section tackles the different works that attempted to put forward an approach for detecting intrusiveness.

3.6.2 Identifying Intrusiveness

In the work presented by Hinckley and Horvitz (2001), intrusiveness or interruptibility as the authors preferred to call it, is measured using the likelihood of the user to respond

[17] http://dictionary.cambridge.org/

to phone calls computed using sensors embedded within the user's mobile device. These sensors were able to detect the user's proximity regarding the device : *The user holds the device*; *The device is close to the user's head*.

Siewiorek et al. (2003) engineered an application that adjusts the device ring tone according to the user's surroundings inferred from the microphone, the light and the accelerometer. Liu (2004) perceived intrusiveness as an interruption that should be avoided when a user is in a particular emotional state that is depicted by a pedometer and a heart rate monitor. The authors assumed that the user's should not be interrupted or disturbed when the system detects that he/she is "*stressed*" or "*angry*".

As it comes to the recommender systems' domain, several works tackled the intrusiveness notion as a user modelling issue and considered that a non-intrusive approach is an approach that can implicitly figure out the users' preferences and related information (Adomavicius and Kwon 2015, Farinella et al. 2012, Palanivel and Sivakumar 2010).

In the following sections, we present the different approaches that tackled intrusiveness from two different aspects.

3.6.2.1 Non-intrusiveness as implicit user profiling

Lin (2013) described the recommender system he proposed as non-intrusive as he estimates implicitly the user's preferences from the time the user spends in a shop. Melguizo et al. (2007) used the text that was currently written by the user to recommend items that are relevant to the text that was written. They perceive this kind of approach as proactive and non-intrusive as it supports authors in the writing task without asking for their involvement. Pu et al. (2012) designed a location based recommendation system to provide the most possible interesting places to a user when he is moving, according to his implicit preference and physical moving location without the user's providing his preference or query explicitly. They proposed two circle concepts, physical position circle that represents spatial area around the user and virtual preference circle that is a non-spatial area related to user's interests which are based on his historical visiting behaviours. They

assume that their approach is non-intrusive as it does not require the involvement of the user to extract his/her preferences.

Quercia et al. (2010) proposed a system that automatically recommend new friends, track health of friendships and make aware of the user mood by monitoring user's activity with mobile phones including monitoring text messages, phone calls and encounters captured by Bluetooth. They have engineered a new technology for mobile phones that silently keeps track of people's colocation, as well as frequency of voice calls and text messages. They also explored the degree to which the engine can predict users' moods (e.g., happiness, sadness) simply based on their activity. The proposed framework called FriendSensing enables new members of social-networking websites to automatically discover their friends. It also helps existing members to elicit new social relations, as they develop over time. These services uses short-range radio technologies (e.g., Bluetooth) for logging encounters and rely on colocation records to elicit relevant encounters and to arrange them into a weighted social network for recommending friends.

Lee and Park (2007b) presented a mobile web news recommendation system that incorporates news article attributes and user preferences with regard to categories and news articles. They estimate user preference by aggregating news article importance and recency (calculated by the difference between the time it was posted and the present), and a user segment that is focused on user profiles, reading patterns of news articles, changes in user interest, and usage patterns.

Unfortunately these works and several others that dealt with the intrusiveness concept focused on it as long as it gets to retrieving the user's interests and not regarding the fact that the recommendation itself might disturb the user. Indeed, intrusiveness can also be an issue within the recommendation process and not only in the user's modelling process.

3.6.2.2 Non-intrusiveness as non-disturbing recommendation

The work presented by Bouneffouf (2013) is considered as a pioneer in measuring the intrusiveness as a phase in which we assess the risk of disturbing the user before rec-

ommending. They consider a situation as a triplet composed of *location*, *time* and *the user's agenda activity*. They define a "*critical*" or "*risky*" situation as a situation in which a user does not want to be disturbed. For each situation, they compute a risk score that depends on the risk-level of the concept describing the user's activity depicted from his/her agenda. They assume that a situation is deemed risky if its risk score exceeds a pre-defined threshold.

Bedi and Agarwal (2012) integrated a situation assessment phase in their approach for recommending restaurants, in which they use fuzzy logic as an inference technique that depends on distance, time, budget and reachability to assess the context level of a given situation. They predefine the fuzzy sets for the context level and for each attribute as, for example:

```
Distance={Near,Moderate,Far}
Context-level={Low,Medium,High}
```

The function gathering the context attributes and the context-level is represented as rules; meaning that the context level is inferred depending on the values of the attributes. Example:

```
IF(Distance IS 'Near') AND (Time IS 'In-Time') AND (Budget IS 'Affordable')
AND (Reachability IS 'High') THEN Context-level IS 'High'
```

Dali Betzalel et al. (2015) considered intrusiveness in a recommendation approach as a classification problem which aims at identifying whether a given context is "*good*" or "*bad*" to trigger the recommendation process. They collected mobile data over a three weeks user study in order to learn the classification model.

Nevertheless, even when some works tried to deal with the intrusiveness issue, they always tend to look at the surroundings of the user forgetting that the big amount of applications embedded in the user's device could be the issue itself.

3.7 Conclusion

Context-aware proactive recommender systems are considered as systems that are able to provide a solution to the information overload problem. In this chapter, we presented the various types of recommender systems and exposed the techniques used within. We also introduced the different evaluation formalisms, that are put forward for such systems, and the limitations that hinder their performances. At this level, we noted the absence of a standard evaluation framework that allows a comparative evaluation of recommender systems. Indeed, the proposed evaluation methodologies are geared to specific domains and targeted technologies that are not usually reusable.

Furthermore, we exposed the intrusiveness aspect that is not sufficiently considered within the recommendation process and we presented the different methods that attempt to identify it.

Part III

A Proactive and Non-Intrusive Recommendation Approach

Chapter 4

A Mobile Situation-Aware Proactive Recommendation Approach

4.1 Introduction

There are several context aware systems that attempted to meet the challenge of providing the right information at the right time without the interference of the user in a mobile environment. However, this requires an efficient modelling of the dimensions of the context and especially the modelling of the user profile. Indeed, as mentioned by Mizzaro and Vassena (2011), several dimensions of context, such as location, time, users activities, needs, resources in the nearbies, light, noise, movement, etc., have to be managed and represented which requires a big amount of information, are time consuming and generate complex context models. On the other hand, context models integrating few dimensions are unable to figure out the whole user context.

In this chapter, we describe a proactive context-aware recommendation approach that integrates the modelling of a situational user profile and the definition of an aggregation frame for contextual dimensions combination.

4.2 Situation-based proactive recommendation

We propose an approach for proactive situation-aware recommendation that covers several domains and recommends the right item when it is most needed without waiting for the user to initiate any interaction or activity with his/her device.

This approach integrates information related to a user gathered from his/her social networking accounts, Facebook[1] in our case, along with mobile technologies in order to proactively recommend relevant information to the user. Therefore, the recommendation process entails a context model that figures out what and when to recommend the relevant information (news, movies, a place to visit, a restaurant, ...) to the user.

A general overview of the approach is described in Figure 4.1.



Figure 4.1: A general overview of the proposed approach

As illustrated in figure 4.1, the user's related information along with low-level information, extracted from the user's mobile device, is leveraged in order to infer the user's situation.

[1] https://www.facebook.com/

This latter aims at identifying the user's need in information (POI, News, ...).

4.2.1 Context modelling and acquisition

The approach that we put forward aims at recommending relevant items that match a user's situation without waiting for the user to initiate an activity. We consider that the user's daily routine is represented as a pack of situations described by the spatio-temporal dimensions and the user's profile and which are organized within a knowledge database that reflects a specific information category.
Thus, context is defined by three dimensions as : $context = profile, location, time$

- The user Profile : user's related information and interests
- Location: the user's position extracted by GPS coordinates
- Time: numerical or temporal labels (morning, evening, ...)

These dimensions are instantiated using the sensors embedded in the user's mobile device in order to capture the context.
The following sections define the different components of the context modelling process.

4.2.1.1 The user's profile

The user profile (UP) stands for the user's interests related to specific general categories C defined by the following set:
$C \in$ *{Restaurants, News, Traffic information,POI(points of interests), Coffee shops, recipe's ideas, pubs, TV program, Concerts, Movies, Hotels, Books, Gift ideas}*
These predefined categories are enriched progressively using the user's interaction with the application.
A category is represented as a set of weighted terms associated to the user's interests regarding this particular category : $C_i = \{t_j^{(i)}, w_j^{(i)}\}; j = 1..m$
The terms extracted from the user's social networking account and specifically from information related to the pages that the user "*likes*", along with his search query terms

depicted from the browsing log, are leveraged to determine the user's preferences. Indeed, statistics[2] indicated that social networks are commonly used during daily life events like going to the movies and shopping. The survey showed that respondents are about 4-5 times more likely to use Facebook and Twitter in such activities (see Figure 4.2).

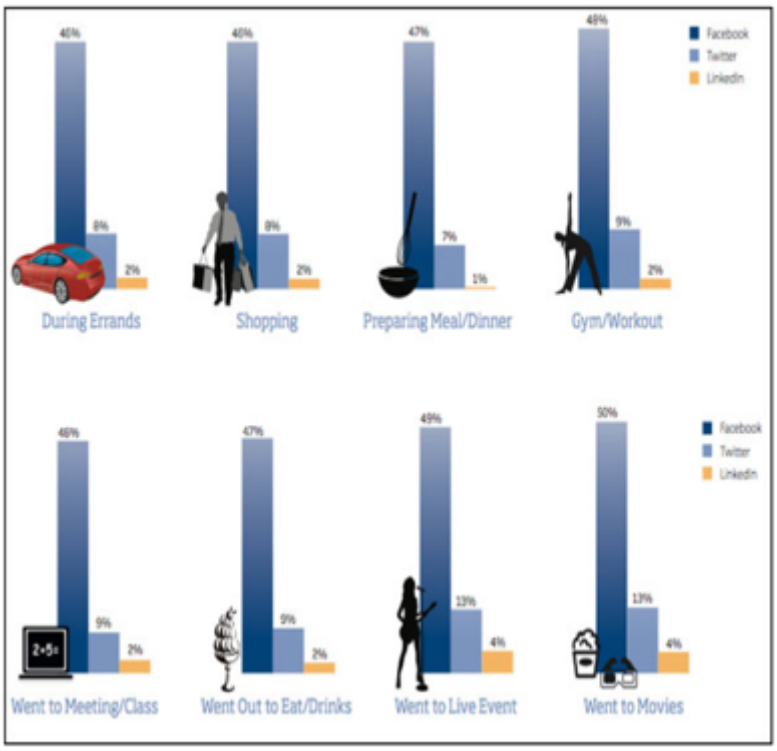


Figure 4.2: Activities through social networks

Then, the KNN data classification algorithm is employed in order to classify the leveraged information under the categories that we predefined.

The tf-idf measure is used to calculate the term weights as follows :

$$TFIDF(t_k, c_j) = TF(t_k, c_j).log\frac{n}{n_k} \tag{4.1}$$

Where n is the number of categories and n_k is the number of categories in which the term appears at least once.

$$TF(t_k, c_j) = \frac{f_{k,j}}{max_z f_{z,j}} \tag{4.2}$$

[2]http://www.pewinternet.org/2015/01/09/frequency-of-social-media-use-2/

The maximum is calculated based on the frequency $f_{z,j}$ of all the terms t_z that appear in the category c_j. Then, the term weight is obtained using the cosine normalization that helps to define a weight in a [0,1] interval:

$$W_{k,j} = \frac{TFIDF(t_k, c_j)}{\sqrt{\sum_{k=1}^{|T|} TFIDF(t_k, cj)^2}} \tag{4.3}$$

4.2.1.2 Time

The time conception and measurement flexibility implies different representations of time context information. We consider the hierarchical modelling of time which can define the degree of granularity of the time context information as the following:
$WeekDay = \{Monday,\ Tuesday,\ ...,\ Sunday\}$; $time = \{morning,\ afternoon,\ ...,\ night\}$
Precisely, a day is split into time slots of a certain length that help to determine the information type to recommend. Time is represented according to two levels:

- Time of the day: A daily routine is divided into five periods (morning, midday, afternoon, evening and night) that are framed within 24 hours intervals.
  $DayTime = \{morning\ [07:00,12:00],\ midday\ [12:00,14:00],\ afternoon\ [14:00,18:00],\ evening[18:00,22:00]\}$

- Week day : defined by two main classes that are workdays (Monday to Friday) and rest days (weekend, vacations and public holidays)
  $WeekDay = \{WorkDays\{Monday,\ Tuesday,\ ....,\ Friday\}, RestDays\{weekend,\ vacations,\ public\ holidays\}\}$

4.2.1.3 Location

The GPS (Global Positioning System) sensor integrated in the device defines the user's location according to latitude and longitude.
Dobson (2005) characterized the GPS coordinates in several ways: *Absolute position*; *Relative* (next to, ...); *A Place name*; *A named class* that stands for the place's type,

eg. museum, school, etc. We consider the actual location, that refers to the user's actual location at a given time. Several tools such as Geonames[3] or Foursquare[4] are used to recover the location type.

4.2.2 Information extraction and recommendation

The proactive recommendation approach that we propose covers multiple domains and aims at recommending relevant items that match a user's situation without waiting for the user to demand them or undertake an activity with his/her mobile device. Therefore, we consider that the user's daily routine is represented as a pack of situations organized within a knowledge database, that reflects a specific category of interest described by the the spatio-temporal dimensions' instantiations.

A situation is characterized by three dimensions: *time of the day*, *the weekday* and *the actual location*: $S = (D_t, D_w, D_l)$ where D_t, D_w and D_l are respectively the values for the time of the day, the weekday and the actual location.

We define, according to the user's situation, a particular category of information C to recommend and that is extracted from the following set:

$C \in \{$*Restaurants, News, Traffic information,POI(points of interests), Coffee shops, recipes' ideas, pubs, TV program, Concerts, Movies, hotels, Books, Gift ideas*$\}$.

For example, the situation "*Lunch time*" is inferred by :

D_l : *At work*;

D_w : *Monday*;

$D_t : t \in [12:00, 14:00]$;

For such situation, the category of information that suits the best is "*Restaurant*".

Therefore, we consider that a situation, with its different levels of representation, defines the changing user's need in information. Indeed, a conducted study[5] showed that the first thing that 58% of the interviewed people do in the morning is read email and news.

[3]http://www.geonames.org/

[4]https://fr.foursquare.com/

[5]http://blog.marketo.com/2013/07/email-wanted-dead-or-aliveinfographic.html

Böhmer et al. (2011) performed a study about user's behaviour and have shown that users tend to consult weather and news in the morning (from 7 am to 9 am), sports applications in the afternoon around (2 pm - 5 pm) and read books at late evening (see figure 4.3. We look for the information to recommend by sending a query q, to a social networking

	12am	1am	2am	3am	4am	5am	6am	7am	8am	9am	10am	11am	12pm	1pm	2pm	3pm	4pm	5pm	6pm	7pm	8pm	9pm	10pm	11pm	% of Total Launches	Users
Browser	7.9%	7.7%	7.8%	7.6%	7.3%	7.4%	7.0%	7.9%	8.1%	8.0%	7.7%	7.3%	7.0%	6.9%	6.8%	6.4%	6.6%	6.6%	6.4%	6.6%	7.0%	7.4%	7.5%	7.4%	6.83%	2,398
Comics	4.5%	5.2%	5.4%	5.8%	5.8%	5.6%	5.5%	5.2%	5.4%	5.1%	4.7%	4.3%	4.3%	4.2%	4.2%	4.3%	4.4%	4.0%	4.4%	4.2%	4.1%	4.1%	4.1%	4.4%	4.31%	2,151
Communication	44.9%	41.1%	38.3%	35.4%	31.6%	31.8%	32.7%	34.7%	39.4%	44.8%	49.0%	52.6%	54.8%	55.2%	55.2%	56.1%	55.7%	56.8%	57.1%	56.1%	54.8%	53.3%	52.0%	49.0%	49.50%	2,769
Entertainment	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.02%	126
Finance	0.2%	0.3%	0.3%	0.2%	0.1%	0.1%	0.1%	0.2%	0.3%	0.3%	0.4%	0.5%	0.3%	0.3%	0.4%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.25%	604
Games	3.2%	3.0%	3.0%	2.7%	2.5%	2.3%	2.2%	1.7%	1.9%	1.9%	2.0%	2.1%	2.2%	2.2%	2.2%	2.3%	2.3%	2.2%	2.2%	2.4%	2.7%	3.0%	3.0%	3.2%	2.30%	1,716
Health	0.3%	0.4%	0.4%	0.4%	0.6%	0.6%	0.7%	0.6%	0.4%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.3%	0.2%	0.3%	0.2%	0.3%	0.26%	540
Libraries & Demo	0.4%	0.5%	0.6%	0.7%	0.9%	0.8%	0.7%	0.6%	0.5%	0.4%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.3%	0.3%	0.3%	0.3%	0.30%	1,267
Lifestyle	0.8%	0.9%	1.0%	1.4%	1.3%	1.5%	1.4%	1.4%	1.1%	0.9%	0.6%	0.6%	0.5%	0.5%	0.5%	0.5%	0.6%	0.5%	0.3%	0.4%	0.4%	0.5%	0.5%	0.5%	0.60%	2,132
Multimedia	2.1%	2.1%	2.4%	2.4%	2.7%	2.4%	1.8%	1.8%	1.9%	1.7%	1.8%	2.0%	2.0%	2.0%	2.2%	2.1%	2.2%	2.4%	2.3%	2.3%	2.2%	2.1%	1.9%	2.0%	2.03%	1,713
News	2.6%	2.5%	2.6%	2.5%	2.5%	2.7%	3.3%	3.7%	4.1%	3.6%	3.0%	2.6%	2.5%	2.7%	2.5%	2.4%	2.2%	2.1%	2.3%	2.2%	2.3%	2.2%	2.3%	2.3%	2.46%	1,777
Productivity	3.6%	5.0%	5.0%	5.8%	6.3%	6.5%	6.0%	5.4%	4.8%	5.1%	4.9%	4.3%	4.2%	4.0%	4.0%	3.7%	3.4%	3.4%	3.0%	3.1%	3.1%	3.0%	2.9%	3.2%	3.76%	2,190
Reference	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.6%	0.6%	0.7%	0.5%	0.5%	0.5%	0.4%	0.4%	0.4%	0.4%	0.3%	0.4%	0.4%	0.4%	0.5%	0.5%	0.5%	0.6%	0.47%	903
Settings	1.3%	1.6%	1.5%	1.3%	1.6%	1.2%	1.2%	1.1%	1.3%	1.4%	1.4%	1.4%	1.2%	1.3%	1.2%	1.2%	1.3%	1.1%	1.1%	1.2%	1.2%	1.3%	1.3%	1.4%	1.23%	2,178
Shopping	3.9%	4.5%	3.7%	3.4%	3.2%	3.2%	3.1%	3.0%	3.1%	3.3%	3.2%	3.2%	3.2%	2.8%	2.9%	2.9%	2.7%	2.7%	2.7%	2.7%	2.8%	3.1%	3.6%	3.5%	2.96%	2,556
Social	5.7%	5.0%	4.9%	4.3%	4.2%	4.0%	4.4%	5.1%	5.3%	5.4%	5.2%	5.0%	4.7%	4.8%	4.9%	4.5%	4.5%	4.6%	4.6%	4.9%	5.2%	5.4%	5.8%	5.7%	4.77%	1,902
Sports	0.5%	0.3%	0.3%	0.2%	0.3%	0.3%	0.2%	0.3%	0.3%	0.3%	0.3%	0.4%	0.4%	0.6%	0.7%	0.8%	0.9%	0.8%	0.6%	0.6%	0.7%	0.8%	0.7%	0.7%	0.56%	571
Themes	0.2%	0.1%	0.2%	0.3%	0.4%	0.4%	0.4%	0.2%	0.2%	0.2%	0.1%	0.1%	0.1%	0.2%	0.1%	0.1%	0.1%	0.2%	0.1%	0.1%	0.2%	0.1%	0.1%	0.1%	0.14%	249
Tools	10.9%	12.2%	14.6%	17.6%	20.3%	21.5%	21.4%	18.6%	14.7%	10.4%	8.4%	6.8%	6.1%	5.9%	5.9%	5.9%	6.0%	6.1%	5.8%	6.0%	6.3%	6.8%	7.4%	9.1%	7.89%	2,512
Travel	1.4%	1.6%	2.1%	2.2%	2.4%	2.6%	2.2%	1.9%	2.0%	2.1%	2.0%	1.8%	1.9%	1.9%	1.9%	1.8%	2.0%	1.9%	2.2%	2.2%	1.9%	1.7%	1.6%	1.4%	1.86%	1,752
Unknown	4.7%	5.3%	5.1%	5.0%	5.3%	4.4%	5.0%	5.9%	4.6%	4.4%	4.1%	3.8%	3.5%	3.8%	3.7%	3.7%	4.0%	3.6%	3.7%	3.7%	3.7%	3.9%	4.1%	4.5%	3.88%	2,284
Total Launches per Hour	103,604	77,053	53,633	40,332	33,438	30,949	38,161	56,895	83,488	109,550	127,069	142,642	158,876	168,082	169,018	172,935	173,963	179,801	184,012	176,050	163,080	153,835	141,303	123,639		

Figure 4.3: Hourly app usage by category (Böhmer et al. 2011)

service that depends on the type of information that should be recommended.

q is formulated as: *q=(the user's actual location(latitude, longitude), category of interest)*

For instance, we use Feedly[6] which is a news aggregator that compiles news feeds from a variety of online sources, in order to retrieve interesting news, and Foursquare to extract information related to restaurants and points of interests. The purpose behind using social networking services for information retrieval lies in the fact that the information extracted is declared as interesting by other users. Then we will be filtering out from information already considered as interesting those suiting best user's preferences.

[6]http://feedly.com/index.html/discover

We represent the query result by a set of items $I = \{i_1, ..., i_n\}$ that are modelled as weighted terms vectors :
$i_j = \{t_k^j, w_k^j\}; j = 1..n, k = 1..p$
The set I is filtered out by calculating a relevance score, in order to extract the items that match the user's preferences.
The item relevance regarding the category of interest includes two components: the topic and the location relevance.

- Topic relevance

As explained earlier, the user' profile stands for the user's interests that are related to predefined categories (Restaurants, News, POI,...). Each category is represented as a set of weighted terms associated to the user's interests regarding a given category. The topic relevance estimates to which degree an item is related to the user's preferences with respect to the given category and is calculated by the cosine similarity :

$$Topic_{rel}(VC_i, It) = \frac{\sum_{j=1}^{n} VC_i^j * It_j}{\sqrt{\sum_{j=1}^{n}(VC_i^j)^2} * \sqrt{\sum_{j=1}^{n}(It_j)^2}} \tag{4.4}$$

Where:
VC_i: the preferences keywords vector related to category C_i
It: the item keywords vector

- Location relevance

In case where the suggested item is location sensitive, we measure the location relevance by calculating the distance between the two GPS coordinates[7] :
($P1(lat1, long1)$ et $P2(lat2, long2)$) that correspond to the suggested item location and the user's current location.

[7]https://en.wikipedia.org/wiki/Great-circle_distance

The distance between $P1$ and $P2$ is measured as:

$$Location_{rel}(P1, P2) = R * c \tag{4.5}$$

Where:
R: The earth radius=6,371Km

$$c = 2 * atan2(\sqrt{a}, \sqrt{(1-a)}) \tag{4.6}$$

$$a = sin^2((lat2 - lat1)/2) + cos(lat1) * cos(lat2) * sin^2((long2 - long1)/2) \tag{4.7}$$

The accessibility formula gives two GPS coordinates in terms of meters.
Thus, we normalize the scores obtained so they can fit into a 0-1 delimited interval [0..1] where 0 refers to a non relevant location and 1 stands for a perfect location relevance.
The normalization formula is described as the following:

$$norm_x_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{4.8}$$

Where x_i stands for the distance separating the user's current location and the suggested item's location.

- The overall relevance calculation

The overall relevance of the item subject to suggestion is computed using a linear combination of the item topic relevance and the item accessibility :

$$Rel = \alpha * Topic_{rel}(C_i, It) + (1 - \alpha) * Location_{rel}(user's_location, item_location) \tag{4.9}$$

In order to set the α coefficient, we performed several tests in which we varied the α coefficient in order to get the best combination relevance. The best overall relevance was

obtained when α is set to 0,6.

The following algorithm summarizes the recommendation process :

Algorithm 1 The recommendation process

Require: *Profile*$\{\{C_i, w_i\}^{i=1..n}\}$
  $C_i = \{(t_j^i, w_j^i); j = 1..m\}$
  Situation $\{D_l, D_t, D_w\}$
  Situation Knowledge Database (KB)
  Case of D_t
  for *each item type* **do**
    $I \leftarrow$ get(*service*,*type*,D_l)
    for *each* $i \in \{I\}$ **do**
      Compute topic-relevance of i
      if *item type is accessibility sensitive* **then**
        compute geo-relevance of i
      end if
      Compute the overall relevance of i :
      $R(i) \leftarrow f(topic_relevance(i), geo_relevance(i))$
    end for
  end for

4.3 Experiments

4.3.1 Experimental framework

4.3.1.1 The TREC Contexual Suggestion Track

The TREC Contexual Suggestion Track (CST) offers an evaluation platform for search techniques that depend highly on the context and the user interests.

The 2012 CST was the only track that was relevant to our approach as it included all the dimensions we use.

The task's input includes a set of suggested venues that were evaluated by a set of users on a five-point scale based on how much a user might find a venue interesting.

The suggested venues are represented by an id, a title, a description and a URL pointing to the venue's website :

```
<example number="2">
<title>Milagro Restaurant</title>
<description>Toronto's first high end Mexican cantina is located in the
heart of the entertainment district.  Milagro is an original concept
created by Andres and Arturo Anhalt, brothers born and raised in Mexico
City, inspired by the traditional Mexican cantinas of the golden era where
food is outstanding, beverages are well served, service is warm and the
atmosphere is relaxed.</description>
<url>http://www.milagrorestaurant.com</url>
</example>
```

The task also includes a set of contexts that correspond to a particular location characterized by *a city*, *day of the week*, *time of day*, and *season*.
For instance, a context might be *Los Angeles, California, on a weekday morning in the fall*.
The context representation :

```
<context number="5">
<city>Los Angeles</city>
<state>CA</state>
<lat>34.05223</lat>
<long>-118.24368</long>
<day>weekday</day>
<time>evening</time>
<season>spring</season>
</context>
```

The two sets are used to leverage the users' preferences regarding the kind of venues

the users would like to visit.

The user's profile is represented by an id and two scores evaluating every venue the user has visited. The "*initial*" score indicates the user's preference regarding the venue's title and description. The "*final*" score is given according to the venue's website content:

```
<profile number="1">
<example number="1" initial="-1" final="0">
<example number="2" initial="1" final="1">
...
<example number="49" initial="1" final="1">
</profile>
```

For each profile/context pairing, it is required to generate a list of fifty venues that are deemed appropriate to the user's profile based on his/her preferences and to the context. Every venue should be characterized by a description, a title and a URL.

4.3.1.2 Evaluation metrics

A survey was conducted in order to collect the users' judgements regarding the venues they were suggested for every context. The suggestions of venues that were submitted by every run participating at the track were judged regarding two components: the profile relevance and the context relevance.

There were two measures used for the TREC track evaluation : precision at rank 5 (P@5) and the mean reciprocal rank (MRR) up to rank 5 (MRR@5).

The P@5 is the proportion of the top 5 relevant suggested places and is calculated as :

$$P@5 = \frac{r}{5} \tag{4.10}$$

Where r stands for the number of relevant suggestions

The MRR@5 is the inverse of the rank of the correct suggestion among the first five suggested places according to a context/profile pair :

$$MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{rank_i} \tag{4.11}$$

where n is the number of context/profile pairs and $rank_i$ refers to the first relevant suggested place rank among the first five suggested places.
There were 10 scores computed by the P@5 and the MRR@5 measures. These latter stand for :

- The geographical relevance computed by the proportion of the top 5 geographically relevant suggested places (P@5_G) and the inverse of the correct suggestion (deemed geographically relevant) rank among the first five suggested places for a profile/context pair (MRR@5_G)

- The temporal relevance measured by the proportion of the top 5 relevant suggested places according to the time context (P@5_T) and the inverse of the correct suggestion (deemed relevant to the time context) rank among the first five suggested places for a profile/context pair (MRR@5_T)

- The website rating calculated as the proportion of the top 5 relevant suggested places according to their website description (P@5_W) and the inverse of the correct suggestion (deemed relevant according to the venue's website description) rank among the first five suggested places for a profile/context pair (MRR@5_W)

- The geo-temporal relevance computed as the proportion of the top 5 suggested places that were rated as relevant according to the location and time context (P@5_GT) and the inverse of the correct suggestion (deemed relevant to the time and location context) rank among the first five suggested places for a profile/context pair (MRR@5_GT)

- The web-geo-temporal relevance that combines the website, the location and time relevance for P@5 and MRR@5: P@5_WGT, MRR@5_WGT

Then, an over-mean across all profile-context pairs and for each relevance type (geographical, temporal, website,...) was calculated for the approaches (runs) presented within the TREC 2012 track:

$$overmean_{P@5} = \frac{\sum_{p=1}^{np} \sum_{c_x=1}^{nc} P@5}{np \times nc} \tag{4.12}$$

$$overmean_{MRR@5} \frac{\sum_{p=1}^{np} \sum_{c_x=1}^{nc} MRR@5}{np \times nc} \tag{4.13}$$

where np is the number of profiles (34) and nc is the number of provided contexts (50).

4.3.2 Evaluation of our approach

In order to evaluate our approach with the TREC task, we followed the following steps:

4.3.2.1 Profiles building

The profiles are constructed using the list of the suggested venues evaluated by the user. Each suggestion is evaluated according to two ratings: a rating for the venue's title and description and a rating for the venue's website.

The profile should indicate which venues a user likes or does not like. The ratings are fixed on a five-point scale based on how interesting a venue would be for the user if he was visiting the city the venue was in: *4, Strongly interested; 3, Interested; 2, Neutral; 1, Disinterested; 0, Strongly disinterested; -1, Website didn't load or no rating was given.*

We consider that a user's profile is organised into predefined time related categories. For example, if it is a *sunny morning*, we recommend to the user outdoor activities such as parks. If it is a *rainy evening*, we suggest indoor activities.

Specifically, the time related categories are predefined as follows:

C $\in$*{Food,Landmarks,Shopping,indoor_activities,outdoor_activities}*

For each profile, these categories are weighted according to the user's preferences extracted

from the example suggestion that he/she evaluated. The weight assigned to a particular category takes into account the two ratings of the suggested venues that were scored by users. One rating for the venue's title and description and the other one is for the venue's website.

$$weight(C) = \frac{\sum_{\forall s \in C} R^s_{td} + R^s_w}{N_p} \tag{4.14}$$

Where:

$\forall s \in C$: for each suggestion s belonging to this category C

R^s_{td}: The venue's title and description rating

R^s_w: The venue's website rating

N_p: the number of suggestions belonging to this category

For example, the profile of a user who likes to go to restaurants and coffee shops and who prefers to visit landmarks of a city rather than shopping, can be represented as :

$profile_u$=*(Food,0.4); (Indoor_activities,0.1); Outdoor_activities,0.2); (Shopping,0); (Landmarks,0.3)*

A profile is then expressed as a set of weighted categories under which there are terms set related to the liked suggestions :

$profile = \{C_i, w_i\}; i = 1..n$

4.3.2.2 Contexts processing

As we explained earlier, we consider each context as a situation characterized by a particular category of interest. For example, we believe that it would be more convenient to suggest to users to go to see a movie or to visit theaters in the evening.

We proceed with gathering the venues related to each context using three geo-based services: Google Places[8], Foursquare and Yelp[9].

The task requires that we suggest a set of venues according the profiles' preferences and to the contextual information provided and which entails the user's actual location expressed

[8]https://developers.google.com/places/

[9]https://www.yelp.com

by *geographic coordinates(latitude,longitude)*, *time of the day (morning,midday,afternoon, evening)* and *the season*.

In order to extract possible interesting venues for each context, we collect venues from Google Places API, Foursquare and Yelp for each category. Thus, we send a query to these geo-based services modeled as

Query={(latitude,longitude),perimeter of the search,category}

The query's results stands for a set of venues represented as:

venue={name,url,description,accessibility,category}

accessibility is the distance between the two GPS coordinates representing the venue and the specified location (see formula 4.5).

Let us take the following example:

- Given the following context :

```
<context number="5">
<city>Los Angeles</city>
<state>CA</state>
<lat>34.05223</lat>
<long>-118.24368</long>
<day>weekday</day>
<time>evening</time>
<season>winter</season>
</context>
```

- For every category $C_i \in$*{Food,Landmarks,Shopping,Indoor_activities,Outdoor_activities}*, we send the query $q_i = \{(34.05223, -118.24368), 800m, C_i\}$.

- An example of returned venues could be :

```
<Nickel Diner;http://nickeldiner.com/;"We're a small diner Striving to
provide a fun, comfortable place for the community to meet, while feasting
on a local home cooked meal.";1043m;food>

<The Theatre at the Ace;http://213nightlife.com/sevengrand;"The Theatre at
Ace Hotel is our loving reanimation of one of the city's most remarkable
gems.  It's a delicately restored, 1,600-seat cathedral to the arts, with
a three-story, 2,300 square foot grand lobby, an ornate open balcony and
a vaulted ceiling dotted with thousands of mirrors that glimmer like tiny
stars.";586m;indoor-activities>
```

As we can notice, we only retrieve venues related to *restaurants* and *indoor_activities* since we consider that is not appropriate to suggest to a tourist to undertake outdoor activities , shopping or visiting landmarks on a weekday evening in winter.
Finally, we obtain venues sets for each context classified under the specifies categories :
$Context_{venues}$ ={C_1{$venue_{11}$, ...,$venue_{1n}$},...,C_i{$venue_{i1}$, ...,$venue_{im}$}}

4.3.2.3 The profile/suggestions matching phase

As we explained earlier, after extracting venues according to the given contexts, we obtain different venue sets regarding each context/category pair. Then, for every $profile_i/context_j$ pair, we generate from the obtained venues sets that are related to $context_i$, those suiting best $profile_i$ preferences using the relevance scores formulas explained earlier (see Formula 4.4, 4.5 and 4.8).
The task requires that we suggest fifty venues for every *profile* P_i/*context* Cx_j.
Every profile (i.e. user) P_i is represented by the weighted predefined categories : $P_i = \{C_k, w_k\}$. For each category C_k, we extract the set of venues V_k that were retrieved for context Cx_j and that are deemed appropriate to the preferences of P_i.

Then, since we only get to suggest for a profile P_i fifty venues for each context, we only extract from V_k, $50 \times w_j$ venues for each category C_k.

Let us return to the previous example of the user who has a profile :

$profile_u$={(Food,0.4); (Indoor_activities,0.1); (Outdoor_activities,0.2); (Shopping,0); (Landmarks,0.3)}.

After extracting the venues that suit best the profile preferences according to the context and using the relevance scores, we get for each category the following number of venues :

$profile_u$={(Food,20 venues); (Indoor_activities,5 venues); (Outdoor_activities,10 venues); (Shopping,0 venues); (Landmarks,15 venues)}

The selection process of interesting places for each context/profile pair is summarized as follows:

Algorithm 2 The profiles/suggestions matching

Require: *Profiles* $P\{P_i\{C_j, w_j\}^{j=1..n}\}^{i=1..m}$
  $C_j = \{(t_k^j, w_k^j); k = 1..l\}\}$
  Contexts $Cx\{Cx_p(location, time)^{p=1..c}\}$
  Suggestions $S\{Suggestions_{Cx_p}\{Venue_1, ...Venue_v\}^{k=1..c}\}$
  for $P_i \in P$ **do**
    for $Cx_p \in Cx$ **do**
      for $Venue \in Suggestions_{Cx_p}$ **do**
        Compute overall relevance of Venue (Formula 4.3)
      end for
      $S_{P_i} \leftarrow Rank(Suggestions_{Cx_p}, 50)$
      $SendS_{P_i}toP_i$
    end for
  end for

4.3.3 Participants' runs

The TREC Contextual Suggestion Track entails four baselines. The first one (Waterloo12a) suggests the top fifty venues for each city extracted from Tripadvisor[10]. This latter is also used by the second baseline (Waterloo12b) which only recommends the

[10]https://www.tripadvisor.com/

venues that are deemed relevant to the user's preferences disregarding the other context dimensions. The two other baselines (Baseline A and Baseline B) both use Google Places API[11]. The queries sent to the API were defined according to the time dimension of each context. As an example, for context "*weekend morning*", the two baselines send queries to the Google Places API from 8 am to 10:45 am. The venues were restricted to restaurants, pubs and coffee shops.

There are generally two perspectives according to which the other participants approached the task. The first method consists in fetching all the types of venues, disregarding the time dimension, and then rank them according to the user's preferences and to the context using different techniques like cosine similarity and point-wise Kullback Leibler (Milne et al. 2012, Liu et al. 2012, Yates et al. 2012, Sappelli et al. 2012, Koolen et al. 2012, Yang and Fang 2012). Koolen et al. (2012) used Wikitravel[12], a collaborative travel guide to extract suggestions. These latter were filtered according to their similarity with the user's positive profile depicted from the venues that were previously judged by the user as interesting. The temporal aspect in this approach is ignored.

The runs presented by Hubert and Cabanac (2012) and Fasilkom (2012) tackled the track with a different approach. They extract venues according to categories tailored to each context. They use the vector space model to represent the user's preferences as a positive vector describing the terms related to the venues rated as interesting to visit by the user; and a negative vector entailing the terms attributed to the venues that were judged uninteresting.

As explained in the previous section, in our approach we tried to figure out the type of venues related not only to the time but also to the season dimension of each context, meaning that, for example, we do not suggest to the user to go to the beach in a "*Weekend Evening Winter*" context. Thus, we framed the kind of venues we wanted to extract by attributing a category to the query sent to the geo-based services. Then, we proceeded with a filtering phase that entails an accessibility and a topic relevance scores.

[11] https://developers.google.com/places/?hl=fr
[12] http://wikitravel.org/

The next section describes the results we obtained applying our approach.

4.3.4 Results

As we could not be part of the TREC 2012 contextual suggestion track evaluation and having our suggested venues being evaluated by the users, we extracted from our evaluation data the venues that were fully evaluated in the track for geographical, temporal and website relevance. Since there were some judgements missing for some profile/context pairs, we conducted a user study in which we asked ten participants to rate the venues that were suggested for these profile/context pairs according to the users' profiles. The set of venues judged within the user study includes not only the venues that had their evaluation missing within the TREC evaluation, but also those that were judged in the track. This is actually used, with other parameters, in order to evaluate the judgements coherence of the user study participants.

Therefore, once we have finished with the user study, we calculated the Fleiss KAPPA (Fleiss et al. 2013) coefficient which measures the inter-agreement between the participants. A Kappa coefficient close to 1 indicates a perfect agreement. Table 4.1 presents the different KAPPA coefficients measured regrading the geographical, the temporal and the website judgements for all the participants. Those latter were given information about the users' preferences and the venues (name, description, location, website).

GeoFleiss Kappa	1.000
TempFleiss Kappa	0.775
WebFleiss Kappa	0.853

Table 4.1: The user study inter-agreement

We also tried to compare the participants' judgements scores with those given in the TREC evaluation for some venues in order to figure out if the participants were really able to guess the profiles' preferences and tastes regarding the suggested venues. Therefore, we measured the precision of the scores given by the participants regarding the TREC evaluation scores for the geographical, the temporal and the website aspects.

Table 4.2 presents the precision scores.

Geo_precision	0.909
Temp_precision	0.818
Web_precision	0.727

Table 4.2: The users' study and the TREC evaluations scores matching

The inter-agreement score exceeding the 0.5 threshold and the precision score allow us to deduce that the users' study that we conducted is reliable in order to fill up the remaining missing scores within the TREC evaluation.

Tables 4.3 and 4.4 illustrate the scores we obtained applying our approach compared to the other runs. In order to be fairly compared to the other participants, the calculation of the P@5 and the MRR@5 scores was redone for all the runs. These latter are ranked according to the website, time and location scores (P@5_WGT and MRR@5_WGT).

As we can notice, the proposed approach yields promising results and proves that the classification method of the users' preferences and the suggested venues within time-related specific categories leads to a better contextual relevance.

As illustrated by table 4.3, despite the fact that we scored the highest *P@5* score for the website, time and location combined relevance *P@5_WGT*, however, we did not get the highest P@5 for the geo_temporal combined relevance *P@5_GT*, the geographical relevance *P@5_G* and the temporal relevance *P@5_T*. That can be explained by the fact that the other runs chose to assign a particular attention to the the geo-temporal factors over the user's preferences. Indeed, we note that we score the highest *P@5* for the website relevance *P@5_W* meaning that we were able to single out, with our approach, the venues that are particularly appropriate to the user's profile and not only in terms of location and time. This is also confirmed by table 4.4, whose findings also reveal, through the *MRR@5_W*, that we could efficiently identify the users' preferences. Therefore, we can consider that the topical profile modelling that we defined in our approach along with the parameters set within the venue retrieval phase, such as the radius defining the venue's premises for a context, are effective.

run	P5_WGT	P5_GT	P5_G	P5_T	P5_W
Proposed approach	**0,4125**	**0,4750**	**0,7750**	**0,6625**	**0,7875**
iritSplitV1	0,3375	0,5625	0,8750	0,5750	0,4750
UDInfoCSTc	0,3000	0,6125	0,8625	0,6625	0,4000
gufinal	0,2875	0,7250	0,9250	0,7375	0,4125
ICTCONTEXTRUN2	0,2875	0,5250	0,8625	0,5250	0,3875
guinit	0,2500	0,6500	0,9375	0,6500	0,3625
udelp	0,2375	0,5750	0,9125	0,5875	0,4250
UDInfoCST	0,2375	0,6375	0,8375	0,7250	0,3750
udelnp	0,2125	0,5875	0,9500	0,5875	0,4125
baselineB	0,2000	0,6750	0,8875	0,6875	0,3125
PRISabc	0,2000	0,5750	0,8625	0,5750	0,3500
run02K	0,2000	0,5750	0,9000	0,5875	0,3500
hplcranki	0,1875	0,5875	0,8500	0,6250	0,3750
iritSplitV2	0,1875	0,5125	0,8250	0,5250	0,3375
run01TI	0,1875	0,6000	0,9000	0,6125	0,3875
baselineA	0,1750	0,4375	0,8250	0,5000	0,4500
ICTCONTEXTRUN1	0,1375	0,5500	0,8750	0,5500	0,3125
waterloo12a	0,1375	0,4625	0,9375	0,4625	0,3500
hplcratin	0,1250	0,4625	0,8875	0,4750	0,4250
waterloo12b	0,1250	0,5750	0,8875	0,5750	0,2125
csiroht	0,0750	0,4750	0,8000	0,4875	0,1875
csiroth	0,0750	0,5375	0,8500	0,5500	0,1375
UAmsCS12wSUM	0,0625	0,1750	0,4375	0,3500	0,2750
FASILKOMU01	0,0500	0,5625	0,9250	0,5625	0,0750
UAmsCS12wSUMb	0,0250	0,2000	0,5000	0,3625	0,3000
FASILKOMU02	0,0000	0,5750	0,9000	0,6000	0,0500
watcs12a	0,0000	0,0000	0,0000	0,7000	0,6125
watcs12b	0,0000	0,0000	0,0000	0,5000	0,6625

Table 4.3: The five P@5 measures sorted by WGT

run	M_WGT	M_GT	M_G	M_T	M_W
gufinal	0,4985	0,8438	1.0000	0,8438	0,6823
Proposed Approach	**0,4933**	**0,5771**	**0,8719**	**0,7417**	**0,8021**
iritSplitV2	0,4604	0,7813	0,9063	0,8125	0,5125
UDInfoCSTc	0,4583	0,7500	0,9063	0,8125	0,4896
iritSplitV1	0,4385	0,6969	0,9375	0,6969	0,6167
guinit	0,4187	0,7396	1.0000	0,7396	0,5177
PRISabc	0,4115	0,6927	0,9688	0,6927	0,5229
run02K	0,4104	0,7521	0,8958	0,7521	0,5406
ICTCONTEXTRUN2	0,4010	0,7083	0,9688	0,7083	0,5229
UDInfoCST	0,4010	0,8125	0,8594	0,9688	0,5469
udelnp	0,3594	0,6792	0,9688	0,6792	0,6667
hplcranki	0,3562	0,6979	0,8750	0,7708	0,5260
baselineB	0,3302	0,7813	1.0000	0,7813	0,4990
udelp	0,3281	0,6510	0,9583	0,6510	0,6250
run01TI	0,3177	0,6406	0,8750	0,6406	0,6615
hplcratin	0,3146	0,5781	0,9688	0,5938	0,6583
baselineA	0,3062	0,6354	0,9375	0,6979	0,6510
ICTCONTEXTRUN1	0,2656	0,6615	0,8563	0,6615	0,5104
waterloo12a	0,2469	0,7031	0,9688	0,7031	0,4063
waterloo12b	0,2188	0,6719	0,9063	0,6719	0,3906
UAmsCS12wSUM	0,1688	0,2938	0,5906	0,5094	0,5906
csiroth	0,1302	0,6250	0,8750	0,6250	0,2802
csiroht	0,1063	0,5104	0,8333	0,5229	0,2344
FASILKOMU01	0,0938	0,6615	0,9375	0,6615	0,1146
UAmsCS12wSUMb	0,0833	0,3094	0,6354	0,5333	0,526
FASILKOMU02	0.0000	0,7052	0,9063	0,7365	0,0677
watcs12a	0.0000	0.0000	0.0000	0,8719	0,6823
watcs12b	0.0000	0.0000	0.0000	0,7781	0,6635

Table 4.4: The five MRR@5 measures sorted by WGT

4.4 Implementation : A Mobile Application for Proactive Recommendation

4.4.1 Host of the application

The recommendation approach that we propose has been developed by a Tunisian company called Tunav[13] in order to be launched in the Tunisian market within a project funded by the European Union. The Implemented application is a proactive context-aware recommender system that enables users to get relevant recommendations according to their current situations and tailored to their preferences depicted from their accounts on the social network Facebook.

The Tunav company that we are working with, develops geo-based and GPS tracking systems which offer the access to a large mass of geographic information. These systems are diversifying into different areas of activity including road transport, services, retail, healthcare, tourism, etc. Our collaboration with Tunav implies the development of a context-aware proactive mobile application that entails the user's preferences and covers multiple domain item recommendation.

4.4.2 Application scenario

The mobile application is developed within a client/server model and it is deployed on the server part. The user only gets the visible and the interactive parts of the application on his/her mobile device. According to a time trigger installed on the user's device, an implicit request is sent to the server to be analysed in order to launch the recommendation of the appropriate information. If the information to be recommended is location sensitive (restaurant, POI, ...) the GPS coordinates will also be sent to the server part.

The installation of the application on the user's device implies the launch of a "splash screen".

[13] http://www.tunav.com/

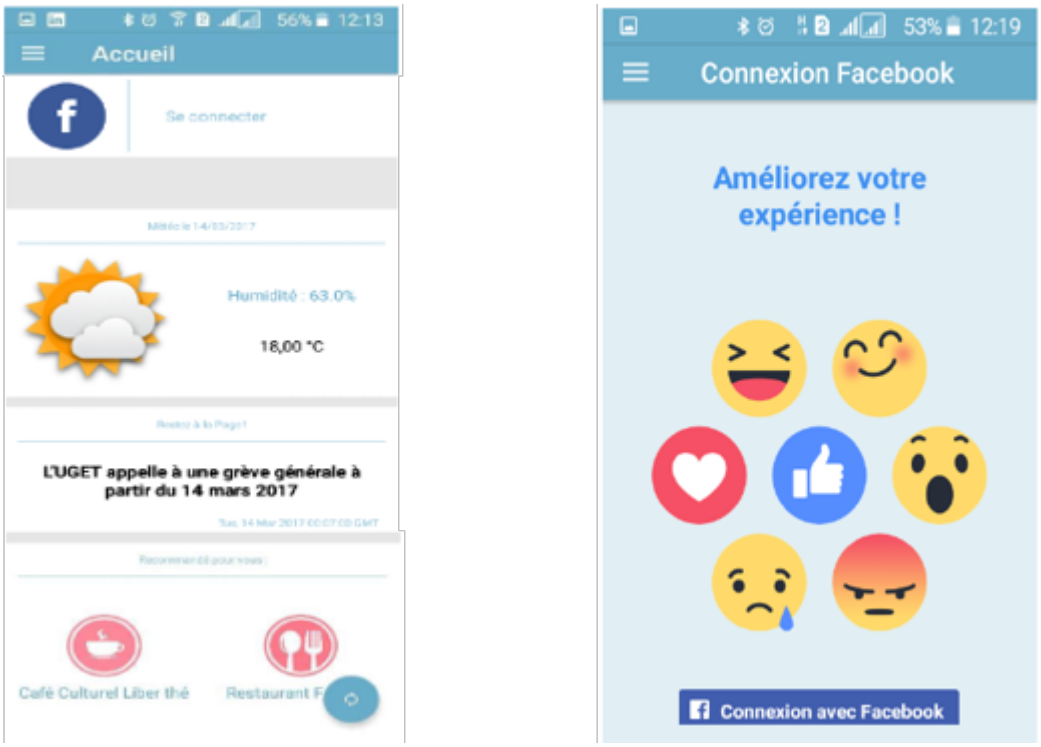


Figure 4.4: The Home and the Facebook fragments

This latter features the "*home fragment*" composed of 4 sub-fragments put together on a scroll view (see Figure 4.4):

- 1^{st}*segment* : *Facebook connection*

  This link allows the user to log into his/her facebook account in order to create his/her user profile for a personalized application usage. Once the user gives permission to access his/her likes, the application collects the required information (see Figure 4.4).

- 2^{nd}*segment* : Weather

  It presents a brief description of the weather according to the user's actual location.

- 3^{rd}*segment* : News

  In this segment, the application displays the headline of a news article that might interest the user.

- $4^{th} segment$: Coffee shops and restaurants

  It presents the nearest coffee shop or restaurant according to the user's location.

The application also presents a menu gathering the different features that can be recommended. This menu allows the user to access interesting information without waiting for an implicit and proactive recommendation (see Figure 4.5).

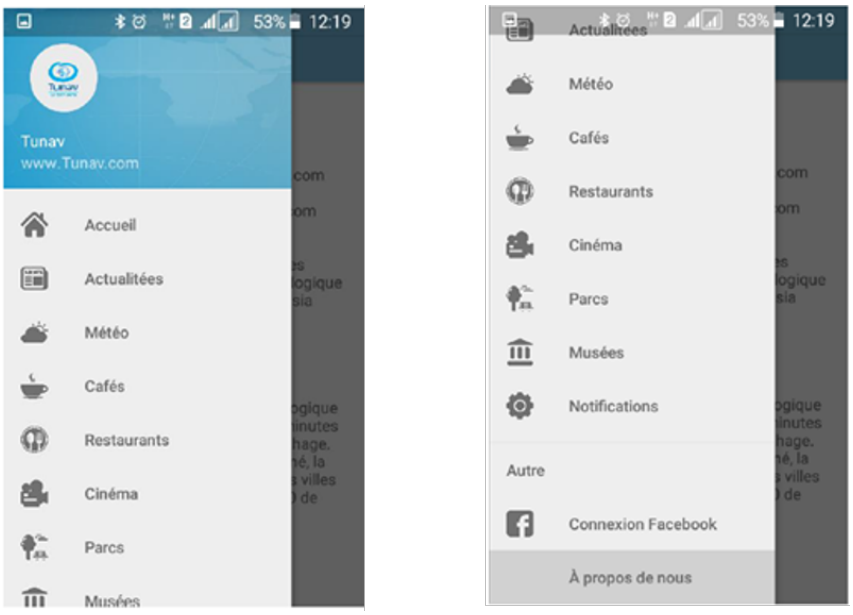


Figure 4.5: Menu fragment

Figure 4.6 and Figure 4.7 illustrate examples of the *news* and the *restaurants* sections. The user can express his/her feedback by clicking on the heart icon down below the screen.

The main purpose of the application is to provide proactive information to the user, besides allowing him/her to check for information manually. Therefore, as we explained in the previous sections, according to given situations, the application implicitly initiate the recommendation process and displays a notification icon entailing a brief description about the recommended information in the mobile's notification bar.
Figure 4.8 illustrates an example of a morning notification about weather and news.
The user has also the possibility of switching on/off notifications about a given information category(see Figure 4.9).



Figure 4.6: News fragment

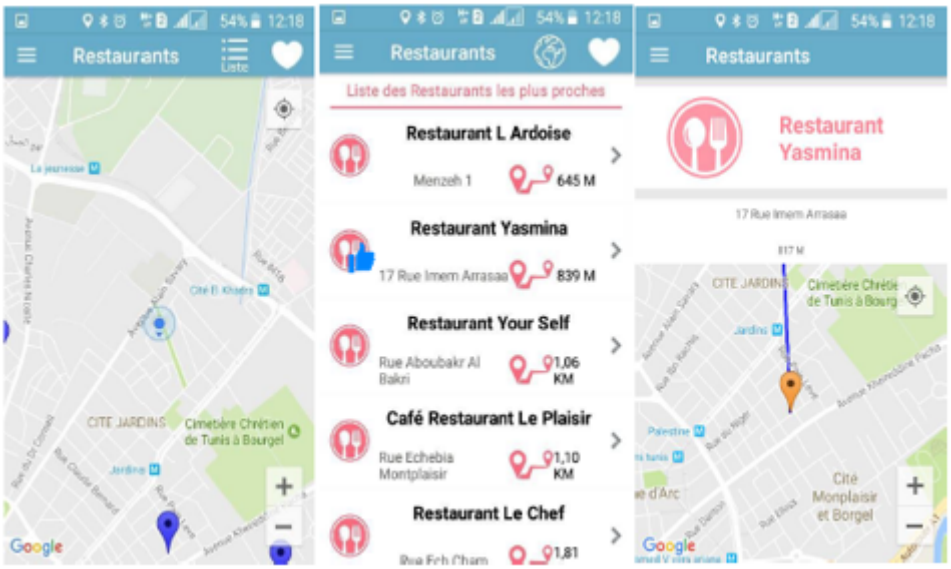


Figure 4.7: Restaurants fragment

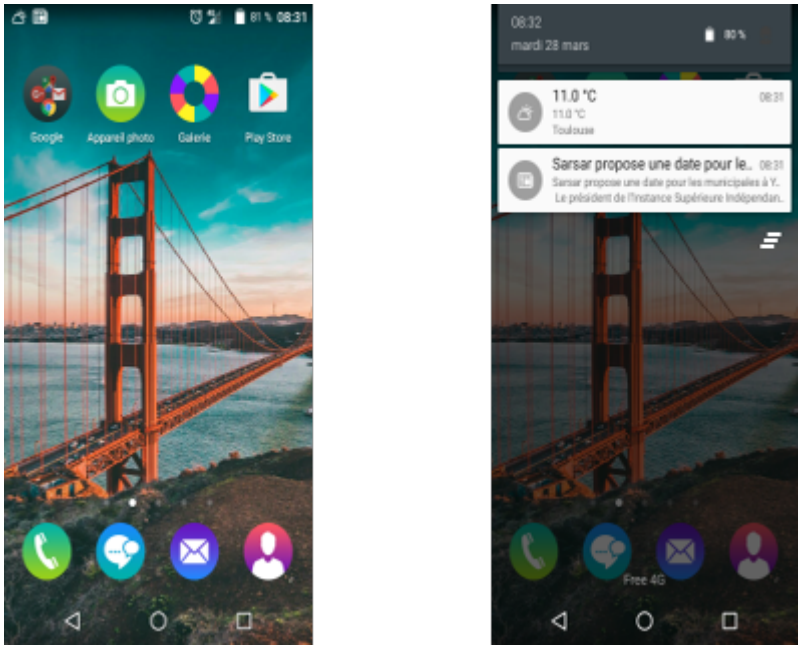


Figure 4.8: Notification example



Figure 4.9: Activating notifications

4.5 Conclusion

In this chapter, we presented our contribution for the development of a situation-aware proactive recommendation approach that consists in recommending information based on the context dimensions and the user's situation without waiting for this latter to initiate any interaction with his/her mobile device. This approach aims at combining the context and the user in the same framework to better characterize the information the user needs at a given situation to improve the recommendation process.

Our approach deals with the challenging issues that we presented in previous chapters, and contributes to the existing works by entailing and combining a proactive non-dependent domain system that covers various domains in the recommendation process and that can help users deal with information overload efficiently by recommending the right item that match users' personal interests at just the right time without waiting for users to initiate any interaction.

We also presented a prototype of a running application that implements the approach that we propose and that was developed by a Tunisian company within a research and innovation support program that is funded by the European Union.

The experiments that we conducted, using the TREC Contextual Suggestion Track, prove that the classification method of the users' preferences within time-related specific categories leads to a better contextual relevance. The results also reveal that the parameters set within the venues retrieval phase defining the venue's premises for a context, such as the radius, are also effective. Nevertheless, this evaluation has only indicted a part of our approach as it only focused on points of interests recommendation and not the multiple domain item recommendation that our approach is based on.

We introduce, in the next chapter, an approach for measuring intrusiveness within the recommendation process.

Chapter 5

Non-Intrusive Recommendations in a Mobile Context

5.1 Introduction

With the advance of mobile technologies and the rapid growth of mobile applications, the user is increasingly confronted with a lot of information. Nowadays, it is not enough for a recommender system to determine what to recommend according to user needs, it also has to tackle the intrusiveness aspect. It is important to include the risk of disturbing the user within the recommendation process. The several works that tried to deal with the intrusiveness concept focused on it as far as it gets to retrieving the user's interests and not regarding the fact that the recommendation itself might disturb the user. Indeed, intrusiveness can also be an issue within the recommendation process and not only in the user's modelling process. Besides, even when some works tackled this issue within the recommendation process, they always tend to look only at the surroundings of the user forgetting that the big amount of applications embedded in the user's mobile device could be the issue itself. In this chapter, we propose an approach for measuring intrusiveness not only in terms of context as generally defined by time and location, but also considering the applications that a user is using at a given situation.

5.2 Situation assessment for non-intrusive recommendations

We propose to integrate an intrusiveness assessment phase that assesses to which degree the recommendation might bother the user. It is about balancing the process of recommendation against intrusive interruptions. In fact, there are different factors that make the user less open to recommendations. As we are working within the framework of mobile devices, we consider that the several embedded applications in a mobile phone such as the camera, the keyboard, the accelerometer, agenda, etc, are good representatives of the user's interaction with his device since they somehow stand for the most used application in a mobile device such as texting messages, chatting, tweeting, browsing or taking selfies and pictures. Indeed, according to a 2015 study[1], 85% of smartphone users spend more than 2 hours a day texting, surfing, talking and tweeting. Thus, we believe that we should take into account the applications that are enabled at a given situation to figure out the user's activity. We adopt a case-based reasoning approach based on the analogous use of past cases to figure out if we could interrupt the user's current activity and send a recommendation.

5.2.1 Situation modelling

We propose to integrate an intrusiveness assessment phase into a situation-aware proactive recommendation approach that covers multiple domains. It aims at recommending relevant items that match a user's situation without waiting for the user to initiate any interaction. The recommendation process is not launched until we assess the intrusiveness level of the situation.

We consider that the user's daily routine is represented as a pack of situations organized within a knowledge database, that reflects a specific category of interest described by the

[1]https://techcrunch.com/2015/06/22/consumers-spend-85-of-time-on-smartphones-in-apps-but-only-5-apps-see-heavy-use/

the spatio-temporal dimensions' instantiations and the user's actual activity.

A situation is characterized by four dimensions: *time of the day*, *the weekday*, *the actual location* and *the user's activity* presented respectively as: $S = (D_t, D_w, D_l, D_a)$.

We define, according to the user's situation, a particular category of information C to recommend and that is extracted from the following set:

$C \in \{$*Restaurants, News, POI, Traffic information, Coffee shops, recipes' ideas, pubs, TV program, Concerts, Movies, hotels, Books, Gift ideas*$\}$.

These predefined categories are enriched progressively using the user's interaction with the application.

For example, the situation "*Lunch time*" is inferred by :

D_l : *At work*;

D_w : *Monday*;

$D_t : t \in [12:00, 14:00]$;

D_a: *the user is taking a break*

For such situation, the category of information that suits the best is "*Restaurant*".

Therefore, we consider that a situation, with its different levels of representation, defines the changing user's need in information. The case-based reasoning approach that we support is based on the use of saved past cases. A user's past case is modelled as $case(premise, value)$:

- PREMISE : contains the situation S_i described by the instantiated dimensions that it entails. The premise is used to measure the similarity between the cases.

- VALUE : integrates 3 parameters, $value(feedback, nb_y, nb_n)$ that refer respectively to the user's feedback associated to the situation S_i, the number of times the user agreed to receive a recommendation at situation S_i and the number of times the user rejected the recommendation at situation S_i.

5.2.2 Retrieval

In order to assess the intrusiveness of a situation, this latter is compared with the past cases. We consider the most similar past situation to figure out if we could interrupt the user's current activity and send a recommendation. For the following sections we denote:
S_c be the current user's situation.
$S_c = \{week_day; time_of_the_day; Current_activity\}$
S the set of past situations stored in the recommendation feedback database.
The system compares S_c with the situations in S in order to figure out the feedback that was given to a similar situation S_p:

$$S = argmax_{S_p \in S} sim(S_c, S_p) \tag{5.1}$$

The similarity between 2 situations takes into account the similarities between the situations' features:

$$sim(S_c, S_p) = \sum_i \alpha_i sim(F_c^i, F_p^i) \tag{5.2}$$

Where F_c^i represents the i^{th} feature of the situation vector S_c (respectively S_p) and $\sum_i \alpha_i = 1$.
In the first experiments that we conducted, we assumed that α_i are equally distributed over the features.
The following sections show how we calculate the respective similarities.

5.2.2.1 Time feature similarity computation

The similarity of the time feature takes into account two levels : time of the day and the week day:

- The week day

  We sequentially enumerate days of the week (*1 for Monday, ..., 7 for Sunday*) in

order to compute the similarity between two week days in terms of proximity as :

$$sim(D_w^c, D_w^p) = 1 - \frac{|D_w^c - D_w^p|}{7} \tag{5.3}$$

- Time of the day

  We choose to divide a daily routine into five periods (*morning*, *midday*, *afternoon*, *evening* and *night*) that are framed within 24 hours intervals.
  $D_t \in \{morning[07:00, 12:00], midday[12:00, 14:00], afternoon[14:00, 18:00], evening[18:00, 22:00]\}$

  In order to calculate the similarity between two time intervals, we rank each period from 1 (*morning*) to 4 (*evening*):

$$sim(D_t^c, D_t^p) = 1 - \frac{|D_t^c - D_t^p|}{4} \tag{5.4}$$

5.2.2.2 The user's activity similarity computation

At a given situation S, the system takes a snapshot of the user's current activity A_c by checking the current enabled application such as *driving, texting messages, chatting, tweeting, browsing or taking pictures*, using the sensors and the applications embedded in the user's mobile device. For example, we can figure out if the user is in a meeting according to his agenda or if the user is taking a picture by checking if the camera is enabled or not.
Thus, the similarity computation of the user's activity related to two situations for this feature is computed as:

$$sim(A_c^c, A_c^p) = \begin{cases} 1 & if \quad A_c^c = A_c^p \\ 1/2 & if \quad A_c^c and A_c^p \text{belongs to the same concept set} \\ 0 & else \end{cases} \tag{5.5}$$

In order to overcome the drawback of syntactic similarity, we consider that two activities could be similar if they belong to the same concept or if they use the same embedded application such as the keypad. In a more formal way, we suppose that the set of activities Ac_i belonging to the same general concept $C_i = \{Ac_1, Ac_2, ..., Ac_n\}$ are deemed partially similar. For instance, if the system detects in a given situation that the user mentioned in his agenda that he has a *meeting*, this latter activity can be considered as approximately similar to other activities like *appointment* or *work*. Figure 5.1 details the proposed user's activity partitioning scheme under general concepts.

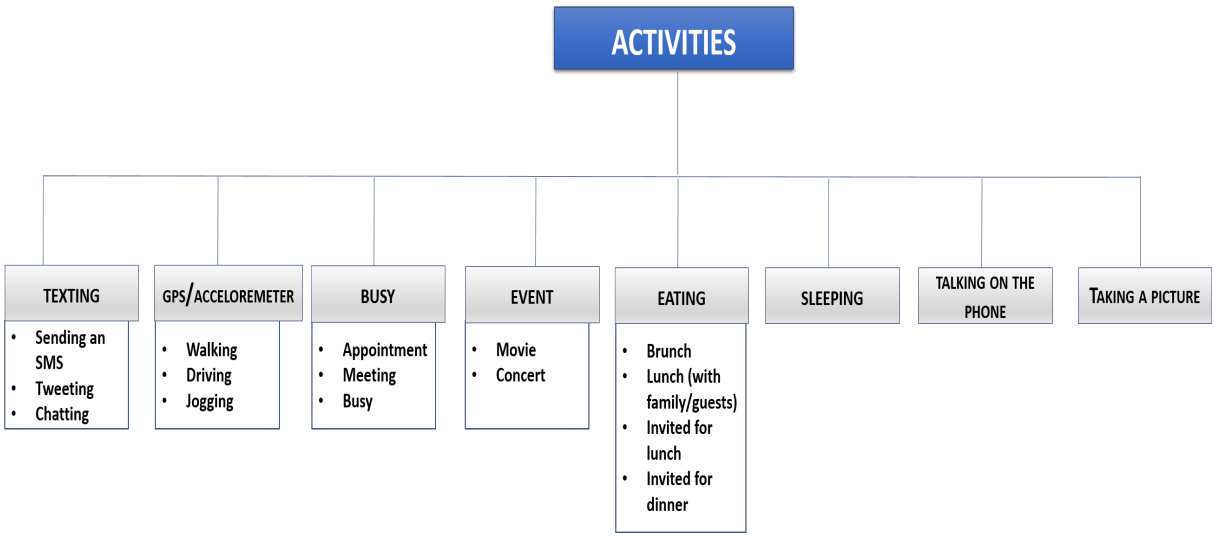


Figure 5.1: Activity partitioning scheme

5.2.3 Reuse

Once we retrieve the most similar past case to the current one, we use the value section that integrates the user's past feedback regarding the similar past situation in order to decide whether we should send a recommendation or not. Meaning that if, for the similar situation, the number of times the user disregarded the notification (nb_n) exceeds the number of times the user agreed to receive a notification (nb_y), we would take that as a "*do not disturb me*" feedback.

In case where we get more than one similar situation (i.e. same similarity score), we

proceed by a voting process by which the most redundant feedback is adopted. We also consider the cold-start problem that arises when there is no similar situation among the past ones. When this occurs, we assume that the recommendation will not bother the user.

5.2.4 Revise

The revision phase consists of recovering the user's feedback regarding the recommendation related to the current situation. The user's click on the recommended information is considered as a "YES" feedback, meaning that the notification did not bother the user. If the user chose to disregard the recommendation by swiping the notification displayed on the device's screen, we take that as a "NO" feedback.
The new feedback of the actual situation may serve for the construction of a new case or update an existing one depending on the similarity score that was previously computed. If we get a perfect match between a past case and the user's actual situation, we accordingly update nb_y or nb_n within the value section (i.e. feedback) of the similar situation. However, the current case will be added to the case database if we do not find an already existing case that corresponds to the actual situation:

Algorithm 3 The revision process

if $(S_c == S_p)$ **then**
  if $feedback_{S_p} =$ "YES" **then**
    $nb_y^{S_p} = nb_y^{S_p} + 1$
  else
    $nb_n^{S_p} = nb_n^{S_p} + 1$
  end if
else
  Add S_c to the case base
end if

5.3 Experiments

Since there is no suitable dataset to experiment the approach we propose, we strived to construct a user study. Indeed, user studies (Bedi and Agarwal 2012, Bouneffouf 2013) are good alternatives for evaluating recommender systems in which users are asked to evaluate recommendations. This kind of evaluation allows a qualitative assessment of the system as surveys can be conducted along with the experiments.

5.3.1 User Study

We automatically generated 100 situations that are characterized by four features : *the week day*, *time of the day*, *the current activity* the user might be doing and *the category of information* that might be recommended (News, POI, restaurant, etc.). The first three features were randomly generated among predefined values that cover most of the activities that a user might perform on his device and that are depictable through the applications installed within the device. Then, according to the situation, the approach we propose assigns automatically the category of information to recommend. For example, a situation can be described to the user as :

```
It is Saturday,Midday and you are doing the following activity:
Taking a picture/selfie
   Would you accept to get a notification :
   YES
   NO
Given this situation, do you think that recommending this type of
information is interesting :  Restaurant
  Is this category of information interesting at this situation?
   YES
   NO
```

Users were asked, given a situation they might be in, if they accept to get a recommendation or not. They were also asked to mention if they consider the information type (News, POI, ...) recommended at that situation as relevant or not.
They also had the possibility to comment on every situation.
We used the crowdflower[2] plateform to run the user study. Figure 5.2 gives an overview about the conducted user study.

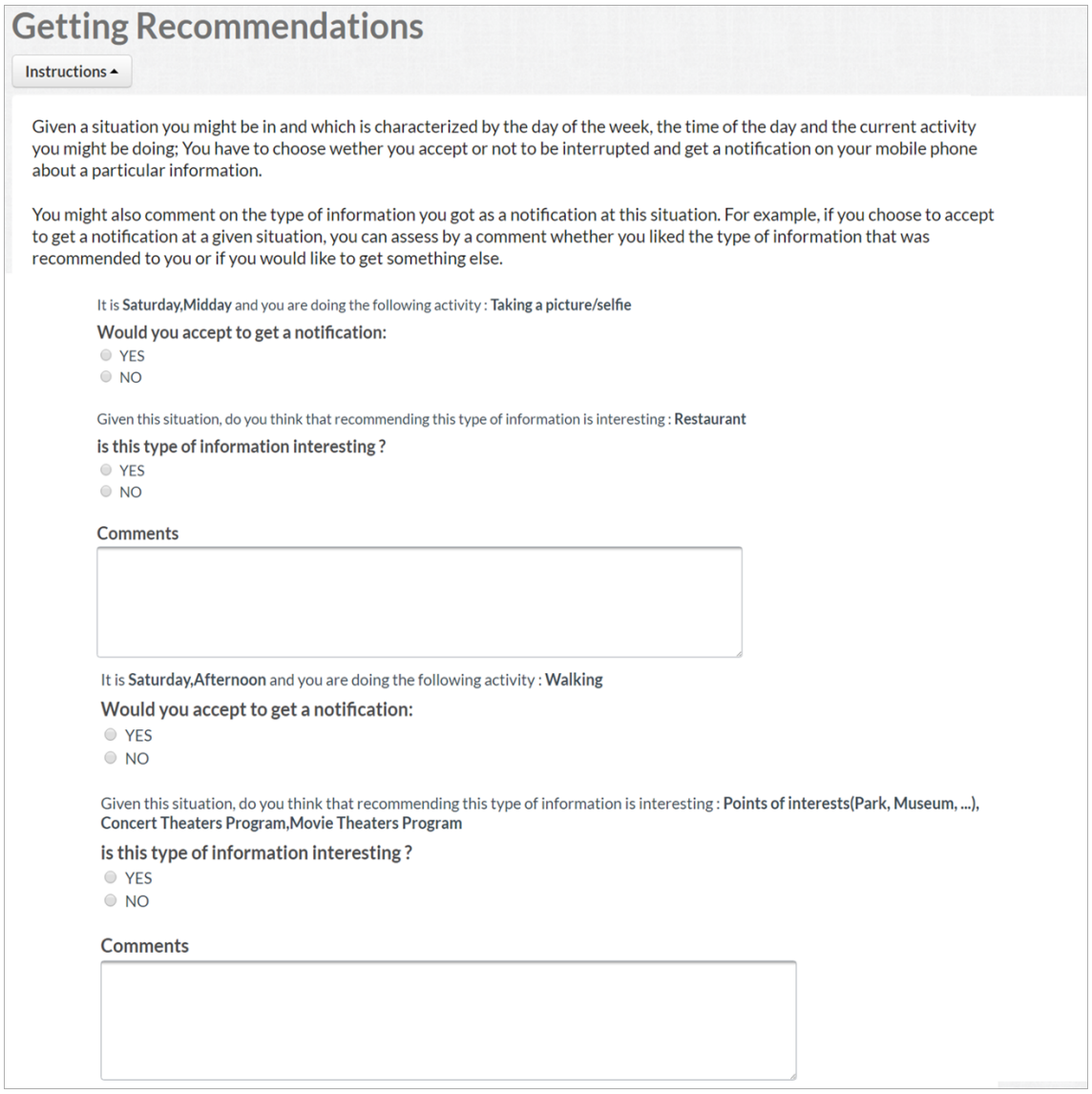


Figure 5.2: The user study overview

[2]https://www.crowdflower.com/

5.3.2 Evaluation

The purpose of this study was to gather real users' judgements about situations that might occur in real life.
Thus, after parsing the collected data, we got about 1500 users who participated to this study.

5.3.2.1 Results

In order to determine the accuracy of the approach in terms of intrusiveness detection, we adopted a cross-validation evaluation that estimates the reliability of a model based on a sampling technique. We run a 10-fold cross-validation test that consists of partitioning, for each user, a sample data that is used as a training set and then use the remaining data for testing. This process is repeated for each user 10 times. Then, we calculated a mean over all users for every possible feature combination.

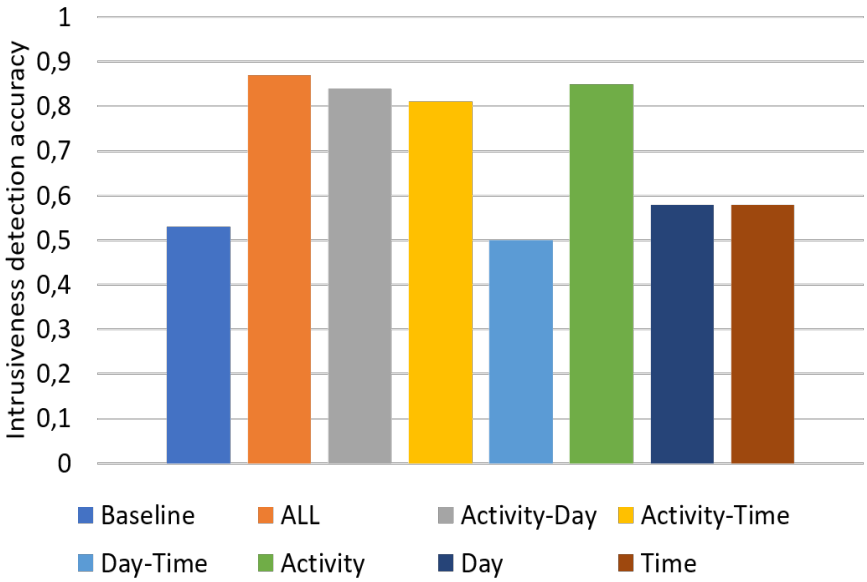


Figure 5.3: The recommendation accuracy using the intrusiveness aspect

As shown in Figure 5.3, the proposed approach, using all the features composing a situation, scores a precision of 87% against 53% for the baseline approach. This latter does not take into account a situation assessment phase, meaning that it sends recommendations without taking into consideration if the user's situation allows to launch the recommendation process or not.

We note that the combinations that entails the activity feature, like *Activity-Day*, *Activity-Time* and *Activity*, scores a high precision that is quite similar to the precision of the approach that makes use of all the features. Then, we can assume that the activity feature is somehow the discriminative attribute for deciding whether a situation is conducive to receive a recommendation or not. Therefore, instead of equally distributing α_i over the features, we should assign to the activity feature a more important weighting coefficient.

As we explained earlier, the category of information to recommend (News, coffee shop, POI, ...) is inferred according to the user's situation. Therefore, we also used this study to put forward the topical relevance of the recommended information regarding the situations that were proposed. For each situation, we measured the proportion of users who rated the recommended information, according to the given situation, as relevant. The approach scored 81% for topic relevance accuracy with an inter-agreement coefficient equals to 0.74 against 58% for the baseline approach that consists at recommending items randomly without taking into consideration the user's situation.

5.3.2.2 Analysis

Given the user study data, we analysed the users' responses and behaviour regarding recommendations according to time and activity. As shown in Figure 5.4, we computed the proportion of users who considered recommendations, in certain activities, as annoying or not. We only put forward the 5 most used applications in a mobile device.

We note that more than 75% of the participants accepted to receive recommendations when tweeting or chatting. This could be explained by the fact that people may want to share with others the recommended information.

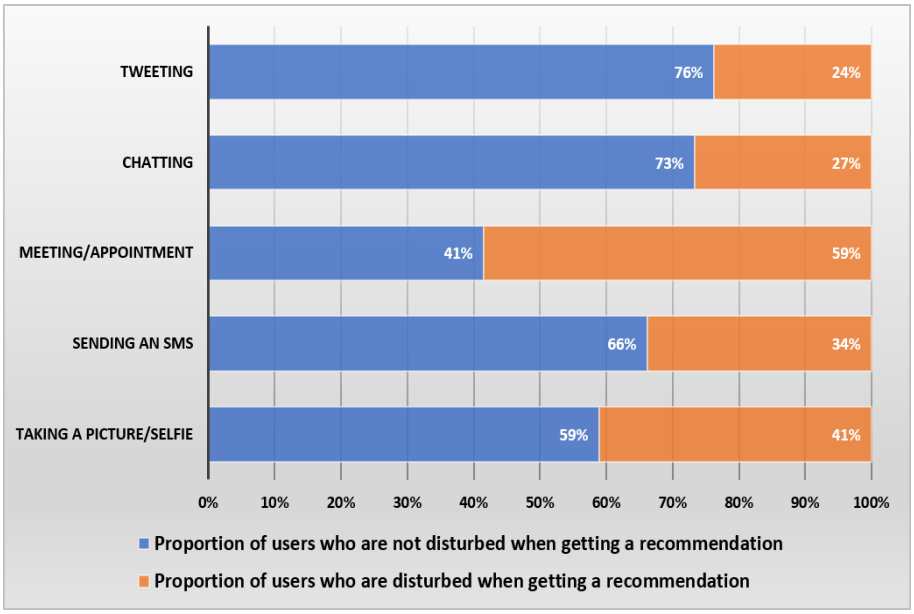


Figure 5.4: The users' behaviour regarding some activities (study conducted with 1500 smartphone users)

We also notice that 59% of the participants against 49% were not disturbed when getting a recommendation while taking a picture which could be somehow interpreted as senseless because we normally expect users to get annoyed if they were interrupted while typing a message or using the device's camera. That is why the case-based reasoning approach, we propose to tackle the intrusiveness aspect, is revealed to be efficient since it considers the specific needs of every user.

We also tried to study the user acceptance regarding receiving notifications according to the time of the day and the day of the week. As expected and as illustrated by Figure 5.5, the two most important peaks to observe happen during breaks and after work. Indeed, it is during these two periods of the day that people have more spare time to spend for activities other than work and chores.

Figure 5.6 shows that the notification acceptance rate follows an escalating pattern starting from the beginning of the week.

People tend to be more receptive to suggestions during weekends.

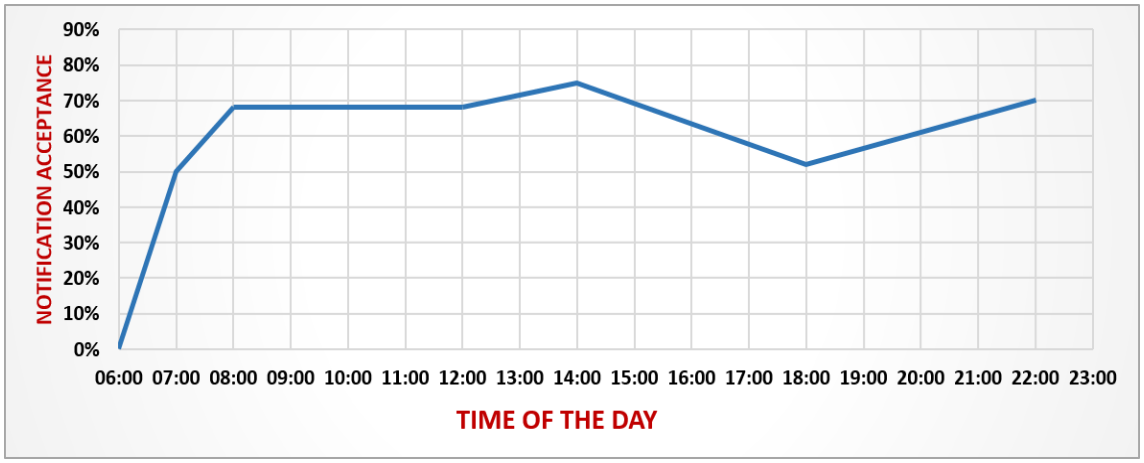


Figure 5.5: The notification acceptance rate according to the time of the day

We can conclude that the user study that we conducted entails a lot of information that can be used for recommender system's evaluation. Indeed, we plan to make this user study available for the RS research community as a dataset for proactive and context-aware RS evaluation. This might help alleviate the datasets shortage and might provide a framework for different approaches to be compared on a same basis.

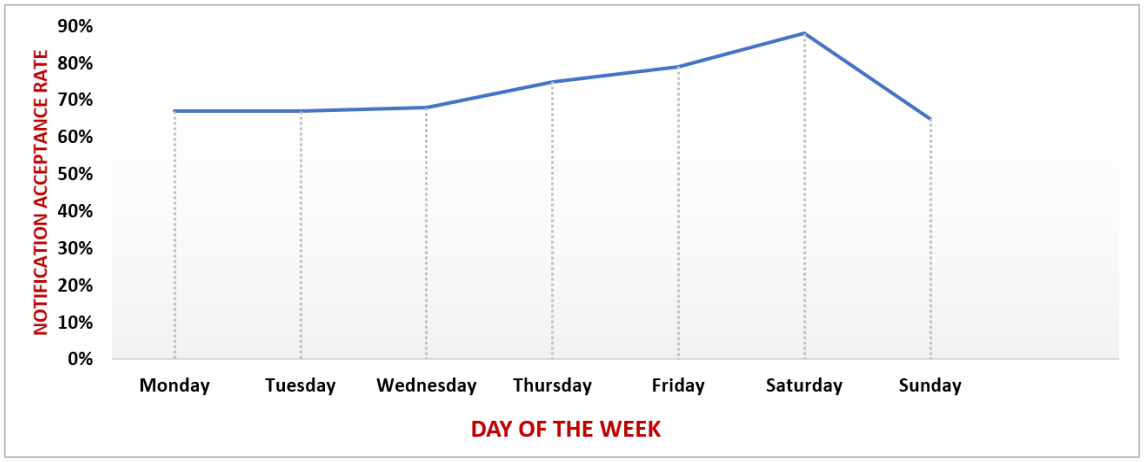


Figure 5.6: The notification acceptance rate according to the day of the week

5.4 Conclusion

In this chapter, we introduced our approach for measuring intrusiveness within the recommendation process. The approach entails a case-based reasoning process that makes use of the user's surroundings and the applications embedded within the user's mobile device in order to assess intrusiveness before recommending. The experiments that we have conducted using a user study yielded promising results. Besides, we were able to construct an evaluation framework based on a user study that can be used to assess context-aware proactive recommender systems effectiveness.

Chapter 6

Conclusion and Research Directions

6.1 Contribution summary

This thesis project proposes a situation-aware proactive recommendation approach in order to recommend synthesized and relevant information without having to wait for the user to initiate an interaction with his/her mobile device. The main thrust of this project tackles just-in-time information, mobile technologies and situation awareness in order to provide users with personalized information tailored to their own needs and preferences at just the right time. Thus, the bibliographical study that we carried out was structured around a presentation of the basic concepts related to context-based systems and proactive recommendation approaches. We presented, in a first chapter, the several works dealing with the various types of context, ranging from the user's personal context to the spatio-temporal context. We exposed the different techniques of context-acquisition, representation and exploitation within situation-aware systems. As reported, we noticed that each system defines its own context model according to the field of application considered. As we are working within the context-aware recommendation domain, we detailed, in a second chapter, how contextual information is employed to boost the performance of recommender systems and particularly the proactive ones. We introduced the different evaluation formalisms, that are put forward for such systems, and the limitations that

hinder their performances. At this level, we noted the absence of a standard evaluation framework that allows a comparative evaluation of recommender systems. Indeed, the proposed evaluation methodologies are geared to specific domains and targeted technologies that are not usually reusable. We also discussed the intrusiveness aspect that is not sufficiently considered within the recommendation process and we presented the different methods that attempted to identify it. We spotted the different issues challenging the context-aware recommendation domain and we were able to frame the proactive recommendation approach that we propose and highlight our contributions regarding existing works, to mention :

1. *A Non-dependent domain system*: we propose to cover various domains in the recommendation process such as news, restaurants, movies, etc.

2. *A Proactive system* : we propose a proactive situation-aware recommender system that can help users deal with information overload problem efficiently by recommending the right item that matches users' personal interests at just the right time without waiting for users to initiate any interaction

3. *A social networks based system*: the genuine interests of the user provided by social networks are of a great help for user profiling. Indeed, social networks provide a wealth of information about the user's interests. Social Networks played a double-edged role within the recommendation approach, as a foundation for user profile modelling and as an information generating resource. We were able to integrate the Facebook API within the mobile application that we developed in order to fetch the user's interests. Besides, we used the Foursquare API for POI extraction.

4. *A Non-intrusive system* : Our approach integrates a situation assessment phase in which we use mobile technologies along with context factors in order to figure out what the different factors that make the user less open to recommendation are.

5. *An evaluation framework* : We were able to construct an evaluation framework based on a user study that can be used to assess context-aware proactive recommender systems effectiveness.

Indeed, we tackled, in a first part, the modelling of a situational user profile and the definition of an aggregation frame for contextual dimensions combination within a proactive recommendation approach that takes advantage from the same context information without encumbering the user's mobile and recommend items related to different domains. The approach, that we presented, integrates user's related information along with mobile technologies in order to proactively recommend relevant information (news, movies, a place to visit, a restaurant, ...) to the user.
We also extended, in a second part, a situation assessment approach in which we consider the intrusiveness aspect within the recommendation process. The situation assessment approach makes use of the user's context and the several applications and sensors embedded within the user's mobile device in order to figure out the situations in which the user might reject recommendations. The experiments that we conducted to assess the situation-aware proactive recommendation approach proposed, using the TREC Contextual Suggestion Track, prove that the situation-aware modelling of the users' preferences within time-related specific categories leads to a better contextual relevance. Besides, the user study that we carried out to evaluate the intrusiveness assessment approach, proved that the case-based reasoning approach we propose to tackle intrusiveness is revealed to be efficient since it considers every user apart. This user study also lays the foundation for an evaluation framework that may help alleviate the datasets shortage and provide a framework for different approaches to be compared on a same basis.

6.2 Perspectives

The various experimental evaluations carried out have shown our contributions' effectiveness regarding a set of models discussed in the literature review.
Furthermore, this dissertation lays the foundation for several perspectives that can be summarized as follows:

1. *Study in-depth the user's surrounding.* To recommend personalized information tailored to the user's situation, we focused on the geo-temporal dimensions and the user's interests. Nevertheless, there are other important aspects that are worth studying. Indeed, we did not consider the user's surrounding in terms of the type of persons that the user might be with, like friends, family or co-workers. We believe that this aspect can enhance not only the relevance of the recommended information but also the effectiveness of the intrusiveness detection. Therefore, we plan to develop an approach that takes into consideration the type of persons the user is being with in order to tailor the information that can be recommended.

2. *Integrating other social networks usage.* We used the user's Facebook account in order to construct the user's profile. However, the user might also possess accounts on other social networks like twitter, that can bring valuable information. Thus, we propose as a perspective, a mashup of other social networks accounts to enrich the user's profile for a better personalized application usage and to allow the user to share the recommended information.

3. *Enhancing the intrusiveness detection.* We intend to integrate into the approach we proposed for intrusiveness detection, a trade-off between the importance of the information to be recommended and the risk of disturbing the user. Indeed, in some situations, even though the user chose not to be disturbed, the recommended information might be worth being interrupted for, such as breaking news or an accident that happened on the user's way home. We believe that such trade-off need to be studied.

4. *Enlarge the evaluation framework.* Despite the fact that we were able to construct an evaluation dataset for situation-aware proactive recommendation in which more than 1500 users with different backgrounds gave their judgements about 100 situations, we believe that this evaluation framework has to be conducted on a real time basis. Therefore, we plan to conduct a user study using the developed application that will give us the chance to gather more information about users' feedbacks and monitor the users' real time usage of the application. Thus, we will be able to study in depth and figure out the other aspects that make the users less open to recommendations.

Bibliography

Lynda Tamine-Lechani, Mohand Boughanem, and Mariam Daoud. Evaluation of contextual information retrieval effectiveness: overview of issues and research. *Knowledge and Information Systems*, 24(1):1–34, 2010.

Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application usage. In *Proceedings of the 13th international conference on Human computer interaction with mobile devices and services*, pages 47–56. ACM, 2011.

Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: Scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 271–280, New York, NY, USA, 2007. ACM.

Robin Burke. Hybrid web recommender systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 377–408. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-72078-2.

Ja-Hwung Su, Hsin-Ho Yeh, P.S. Yu, and V.S. Tseng. Music recommendation using content and context information mining. *Intelligent Systems, IEEE*, 25(1):16–26, Jan 2010. ISSN 1541-1672.

Bradley N. Miller, Istvan Albert, Shyong K. Lam, Joseph A. Konstan, and John Riedl. Movielens unplugged: Experiences with an occasionally connected recommender sys-

tem. In *In Proceedings of ACM 2003 International Conference on Intelligent User Interfaces (IUI'03) (Accepted Poster*, pages 263–266. ACM Press, 2003.

Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, pages 195–204, New York, NY, USA, 2000. ACM. ISBN 1-58113-231-X.

Koji Miyahara and Michael J. Pazzani. Collaborative filtering with the simple bayesian classifier. In *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*, PRICAI'00, pages 679–689, Berlin, Heidelberg, 2000. Springer-Verlag. ISBN 3-540-67925-1.

James Pitkow and Peter Pirolli. Mining longest repeating subsequences to predict world wide web surfing. In *Proceedings of the 2Nd Conference on USENIX Symposium on Internet Technologies and Systems - Volume 2*, USITS'99, pages 13–13, Berkeley, CA, USA, 1999. USENIX Association.

Dmitry Pavlov, Eren Manavoglu, David M. Pennock, and C. Lee Giles. Collaborative filtering with maximum entropy. *IEEE Intelligent Systems*, 19(6):40–48, 2004. ISSN 1541-1672.

Octavian Rolland Arnautu. *Mures : Un système de recommandation de musique*. PhD thesis, Faculty of arts and sciences, University of Montreal, 2012.

E. Vozalis and K. G. Margaritis. Analysis of recommender systems' algorithms. In *The 6th Hellenic European Conference on Computer Mathematics & its Applications (HERCMA), Athens, Greece*, 2003.

Stefano Mizzaro and Luca Vassena. A social approach to context-aware retrieval. *World Wide Web*, 14(4):377–405, 2011.

Tien T. Nguyen and John Riedl. *Predicting Users' Preference from Tag Relevance*, pages 274–280. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

Michal Shmueli-Scheuer, Haggai Roitman, David Carmel, Yosi Mass, and David Konopnicki. Extracting user profiles from large scale data. In *Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud*, pages 4:1–4:6, New York, NY, USA, 2010. ACM.

Wen Li, Carsten Eickhoff, and Arjen P. de Vries. Want a coffee?: Predicting users' trails. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1171–1172, New York, NY, USA, 2012a. ACM.

Qiang Pu, Ahmed Lbath, and Daqing He. Location based recommendation for mobile users using language model and skyline query. *International Journal of Information Technology & Computer Science (IJITCS)*, 4(10):19–28, 2012.

H.J. Lee and Sung Joo Park. Moners: A news recommender for the mobile web. *Expert Systems with Applications*, 32(1):143 – 150, 2007a.

Wouter IJntema, Frank Goossen, Flavius Frasincar, and Frederik Hogenboom. Ontology-based news recommendation. In *Proceedings of the 2010 EDBT/ICDT Workshops*, pages 16:1–16:6, New York, NY, USA, 2010. ACM.

Anatole Gershman, Travis Wolfe, Eugene Fink, and Jaime Carbonell. News personalization using support vector machines. *Proceedings of the Workshop on Enriching Information Retrieval*, 2011.

Abhishek Arora and Preyas Shah. *Personalized News Prediction and Recommendation.* PhD thesis, Stanford University, 2011.

Shweta Athalye. *Recommendation System for News Reader.* PhD thesis, San Jose State University, 2013.

Doina Alexandra Dumitrescu and Simone Santini. Improving novelty in streaming rec-

ommendation using a context model. In *CARS 2012 : ACM RecSys Workshop on Context-Aware Recommender Systems*, 2012.

Andrei Popescu-Belis, Majid Yazdani, Alexandre Nanchen, and Philip N. Garner. A speech-based just-in-time retrieval system using semantic search. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, pages 80–85, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

Susan Dumais, Edward Cutrell, Raman Sarin, and Eric Horvitz. Implicit queries (iq) for contextualized search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 594–594, New York, NY, USA, 2004. ACM.

Margarita Karkali, Dimitris Pontikis, and Michalis Vazirgiannis. Match the news: A firefox extension for real-time news recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1117–1118, New York, NY, USA, 2013. ACM.

Paul Prekop and Mark Burnett. Activities, context and ubiquitous computing. *Comput. Commun.*, 26(11):1168–1176, 2003.

Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. From chatter to headlines: Harnessing the real-time web for personalized news recommendation. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, pages 153–162, New York, NY, USA, 2012. ACM.

Shawn O'Banion, Larry Birnbaum, and Kristian Hammond. Social media-driven news personalization. In *Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the Social Web*, pages 45–52, New York, NY, USA, 2012. ACM.

Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. On using the real-time web for news recommendation & discovery. In *Proceedings of the 20th International*

Conference Companion on World Wide Web, pages 103–104, New York, NY, USA, 2011. ACM.

Bill N Schilit and Marvin M Theimer. Disseminating active map information to mobile hosts. *IEEE network*, 8(5):22–32, 1994.

Peter J Brown, John D Bovey, and Xian Chen. Context-aware applications: from the laboratory to the marketplace. *IEEE personal communications*, 4(5):58–64, 1997.

Nick Ryan, Jason Pascoe, and David Morse. Enhanced reality fieldwork: the context aware archaeological assistant. *Bar International Series*, 750:269–274, 1999.

Anind K Dey. Context-aware computing: The cyberdesk project. In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pages 51–54, 1998.

Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, UK, 1999. Springer-Verlag.

Bryce Allen. Information needs: A person-in-situation approach. In *Proceedings of an International Conference on Information Seeking in Context*, ISIC '96, pages 111–122, London, UK, UK, 1997. Taylor Graham Publishing.

Diane H. Sonnenwald. Exploring the contexts of information behaviour. chapter Perspectives of Human Information Behaviour: Contexts, Situations, Social Networks and Information Horizons, pages 176–190. Taylor Graham Publishing, London, UK, UK, 1999.

Colleen Cool. The concept of situation in information science. *Annual review of information science and technology*, 35:5–42, 2001.

Steve Lawrence. Context in web search. *IEEE Data Engineering Bulletin*, 23:25–32, 2000.

Luz M. Quiroga and Javed Mostafa. An experiment in building profiles in information filtering: The role of context of user relevance feedback. *Inf. Process. Manage.*, 38(5): 671–694, September 2002. ISSN 0306-4573. doi: 10.1016/S0306-4573(01)00058-9.

Peter Ingwersen and Kalervo Järvelin. Information retrieval in context: Irix. In *ACM SIGIR Forum*, volume 39, pages 31–39. ACM, 2005.

Jean-Christophe Bottraud, Gilles Bisson, and Marie-France Bruandet. Expansion de requêtes par apprentissage automatique dans un assistant pour la recherche d'information. In *CORIA*, pages 89–108, 2004.

Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. In *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004.

Gerard Salton and Chu-Sing Yang. On the Specification of Term Values in Automatic Indexing. *Journal of Documentation*, 29:351–372, 1973.

Henry Lieberman. Letizia: An agent that assists web browsing. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 924–929, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

Lynda Tamine, Nesrine Zemirli, and Wahiba Bahsoun. Approche statistique pour la définition du profil d'un utilisateur de système de recherche d'information. *Revue I3 - Information Interaction Intelligence*, 7(1):5–25, 2007.

John Paul Mc Gowan. *A Multiple Model Approach to Personalised Information Access.* PhD thesis, Faculty of Science University College Dublin, 2003.

Ahu Sieg, Bamshad Mobasher, Steve Lytinen, and Robin Burke. Using Concept Hierarchies to Enhance User Queries in Web-Base Information Retrieval. In *Artificial Intelligence and Applications*, 2004.

George Wamamu Musumba and Henry Nyongesa. Context awareness in mobile computing: A review. *International Journal of Machine Learning and Applications*, 2(1): 5–pages, 2013.

Thomas Hofer, Wieland Schwinger, Mario Pichler, Gerhard Leonhartsberger, Josef Altmann, and Werner Retschitzegger. Context-awareness on mobile devices - the hydrogen approach. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS03) - Track 9 - Volume 9*, HICSS 03, Washington, DC, USA, 2003. IEEE Computer Society.

Hyoung R. Kim and Philip K. Chan. Learning implicit user interest hierarchy for context in personalization. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI 03, pages 101–108, New York, NY, USA, 2003. ACM.

Susan Gauch, Jason Chaffee, and Alaxander Pretschner. Ontology-based personalized search and browsing. *Web Intelli. and Agent Sys.*, 1(3-4):219–234, December 2003.

Ahu Sieg, Bamshad Mobasher, and Robin Burke. Web search personalization with ontological user profiles. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM 07, pages 525–534, New York, NY, USA, 2007. ACM.

Fang Liu, Clement Yu, and Weiyi Meng. Personalized web search for improving retrieval effectiveness. *IEEE Trans. on Knowl. and Data Eng.*, 16(1):28–40, January 2004.

Mariam Daoud, Lynda-Tamine Lechani, and Mohand Boughanem. Towards a graph-based user profile modeling for a session-based personalized search. *Knowl. Inf. Syst.*, 21(3):365–398, November 2009.

Kam Yeung. *A context-aware framework for personlised recommendation in mobile environments*. PhD thesis, University of Portsmouth, 2011.

Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 335–336, New York, NY, USA, 2008. ACM.

Cosimo Palmisano, Alexander Tuzhilin, and Michele Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. *IEEE transactions on knowledge and data engineering*, 20(11):1535–1549, 2008.

Harry Chen. *An intelligent broker architecture for pervasive context-aware systems*. PhD thesis, University of Maryland, Baltimore County, 2004.

Tao Gu, Hung Keng Pung, and Da Qing Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and computer applications*, 28(1):1–18, 2005.

Jongyi Hong, Eui-Ho Suh, Junyoung Kim, and SuYeon Kim. Context-aware system for proactive personalized service based on context history. *Expert Systems with Applications*, 36(4):7448–7457, 2009.

Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine*, 10(1):156–167, 2006a.

Martin Berchtold and Michael Beigl. Increased robustness in context detection and reasoning using uncertainty measures: Concept and application. In *European Conference on Ambient Intelligence*, pages 256–266. Springer, 2009.

Diane Kelly and Xin Fu. Eliciting better information need descriptions from users of information search systems. *Information Processing & Management*, 43(1):30–46, 2007.

Xuehua Shen, Bin Tan, and ChengXiang Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR*

conference on Research and development in information retrieval, pages 43–50. ACM, 2005.

Jaime Teevan, Susan T Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456. ACM, 2005.

Magdalini Eirinaki and Michalis Vazirgiannis. Web mining for web personalization. *ACM Transactions on Internet Technology (TOIT)*, 3(1):1–27, 2003.

Bamshad Mobasher. Data mining for web personalization. In *The adaptive web*, pages 90–135. Springer, 2007.

Geoffrey I Webb, Michael J Pazzani, and Daniel Billsus. Machine learning for user modeling. *User modeling and user-adapted interaction*, 11(1):19–29, 2001.

Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE, 1994.

Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 89–101, London, UK, 1999a. Springer-Verlag.

Az Azrinudin Alidin and Fabio Crestani. Context modelling for just-in-time mobile information retrieval. *Pertanika Journal of Science and Technology*, 21(1):227–238, 2013. ISSN 0128-7680.

Paolo Coppola, Vincenzo Della Mea, Luca Di Gaspero, Stefano Mizzaro, Ivan Scagnetto, Andrea Selva, Luca Vassena, Paolo Z, and Paolo Zandegiacomo Rizio. Information filtering and retrieving of context-aware applications within the mobe framework. In

In proceedings of Proceedings of the Workshop on Context-Based Information Retrieval, 2005.

Jay Budzik and Kristian J. Hammond. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, pages 44–51, New York, NY, USA, 2000. ACM.

Michael J Pazzani, Jack Muramatsu, Daniel Billsus, et al. Syskill & webert: Identifying interesting web sites. In *AAAI/IAAI, Vol. 1*, pages 54–61, 1996.

Vishnu Kanth Reddy Challam. *Contextual information retrieval using ontology based user profiles.* PhD thesis, university of Kansas, 2004.

Nikolaos Nanas, Victoria Uren, and Anne De Roeck. Building and applying a concept hierarchy representation of a user profile. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 198–204. ACM, 2003.

Chenxi Lin, Gui-Rong Xue, Hua-Jun Zeng, and Yong Yu. Using probabilistic latent semantic analysis for personalized web search. In *Asia-Pacific Web Conference*, pages 707–717. Springer, 2005.

Ji-Rong Wen, Ni Lao, and Wei-Ying Ma. Probabilistic model for contextual retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 57–63. ACM, 2004.

Giuseppe Amato and Umberto Straccia. User profile modeling and applications to digital libraries. In *International Conference on Theory and Practice of Digital Libraries*, pages 184–197. Springer, 1999.

Joyce Diane Kelly. *Understanding implicit feedback and document preference: A naturalistic user study.* PhD thesis, Rutgers, The State University of New Jersey, 2004.

Robert Armstrong, Dayne Freitag, Thorsten Joachims, Tom Mitchell, et al. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring symposium on Information gathering from Heterogeneous, distributed environments*, pages 6–12, 1995.

Joseph John Rocchio. Relevance feedback in information retrieval. 1971.

Stefano Mizzaro and Carlo Tasso. Ephemeral and persistent personalization in adaptive information access to scholarly publications on the web. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 306–316. Springer, 2002.

Liren Chen and Katia Sycara. Webmate: a personal agent for browsing and searching. In *Proceedings of the second international conference on Autonomous agents*, pages 132–139. ACM, 1998.

Ameni Kacem, Mohand Boughanem, and Rim Faiz. Time-sensitive user profile for optimizing search personlization. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 111–121. Springer, 2014.

Simon Dobson. Leveraging the subtleties of location. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pages 189–193. ACM, 2005.

Uta Christoph, Janno Von Stülpnagel, Karl-Heinz Krempels, and Christoph Terwelp. Context detection on mobile devices. In *Paper präsentiert auf der 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, volume 15, page 17, 2010.

Gerald James Whitrow. *Time in history: Views of time from prehistory to the present day*. Oxford University Press, USA, 1989.

Ourdia Ressad-Bouidghaghen. *Accès contextuel à l'information dans un environnement mobile: approche basée sur l'utilisation d'un profil situationnel de l'utilisateur et d'un*

profil de localisation des requêtes. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2011.

Damianos Gavalas and Michael Kenteris. A web-based pervasive recommendation system for mobile tourist guides. *Personal and Ubiquitous Computing*, 15(7):759–770, 2011.

Pedro G Campos, Fernando Díez, and Iván Cantador. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1-2):67–119, 2014.

Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic, and Erik Duval. Context-aware recommender systems for learning: a survey and future challenges. *IEEE Transactions on Learning Technologies*, 5(4):318–335, 2012.

Yongfeng Zhang, Min Zhang, Yiqun Liu, Chua Tat-Seng, Yi Zhang, and Shaoping Ma. Task-based recommendation on a web-scale. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 827–836. IEEE, 2015.

Bernard J Jansen and Amanda Spink. How are we searching the world wide web? a comparison of nine search engine transaction logs. *Information processing & management*, 42(1):248–263, 2006.

Matthew Richardson. Learning about the world through long-term query logs. *ACM Transactions on the Web (TWEB)*, 2(4):21, 2008.

Salvatore Orlando and Fabrizio Silvestri. Mining query logs. In *European Conference on Information Retrieval*, pages 814–817. Springer, 2009.

Gabriele Tolomei, Salvatore Orlando, and Fabrizio Silvestri. Towards a task-based search and recommender systems. In *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on*, pages 333–336. IEEE, 2010.

Guanling Chen, David Kotz, et al. A survey of context-aware mobile computing research. Technical report, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

Nicky Kern, Bernt Schiele, and Albrecht Schmidt. Multi-sensor activity context detection for wearable computing. In *European Symposium on Ambient Intelligence*, pages 220–232. Springer, 2003a.

Ling Bao and Stephen S Intille. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing*, pages 1–17. Springer, 2004.

MJ Mathie, ACF Coster, NH Lovell, and BG Celler. Detection of daily physical activities using a triaxial accelerometer. *Medical and Biological Engineering and Computing*, 41 (3):296–301, 2003.

Toru Nakata. Recognizing human activities in video by multi-resolutional optical flows. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 1793–1798. IEEE, 2006.

Yongwon Cho, Yunyoung Nam, Yoo-Joo Choi, and We-Duke Cho. Smartbuckle: human activity recognition using a 3-axis accelerometer and a wearable camera. In *Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, page 7. ACM, 2008.

Djallel Bouneffouf. *DRARS, A Dynamic Risk-Aware Recommender System*. PhD thesis, Institut National des Télécommunications, 2013.

Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In *International Symposium on Handheld and Ubiquitous Computing*, pages 89–101. Springer, 1999b.

Patrick Morris Luley, Lucas Paletta, and Alexander Almer. Visual object detection from mobile phone imagery for context awareness. In *Proceedings of the 7th international*

conference on Human computer interaction with mobile devices & services, pages 385–386. ACM, 2005.

Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine*, 10(1):156–167, 2006b.

Nicky Kern, Bernt Schiele, and Albrecht Schmidt. Multi-sensor activity context detection for wearable computing. In *European Symposium on Ambient Intelligence*, pages 220–232. Springer, 2003b.

Ling Ma, Dan Smith, and Ben Milner. Environmental noise classification for context-aware applications. In *International Conference on Database and Expert Systems Applications*, pages 360–370. Springer, 2003.

Vivian Genaro Motti and Kelly Caine. Usersâ€™ privacy concerns about wearables. In *International Conference on Financial Cryptography and Data Security*, pages 231–244. Springer, 2015.

John McCarthy and Patrick J Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Readings in artificial intelligence*, pages 431–450, 1969.

John McCarthy. *Situations, Actions, and Causal Laws*. Stanford Artificial Intelligence Projec). MIT Press, Cambridge, 1963.

Raymond Reiter. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT press, 2001.

Mica R Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

Stephen S Yau, Dazhi Huang, Haishan Gong, and Siddharth Seth. Development and runtime support for situation-aware application software in ubiquitous computing envi-

ronments. In *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, pages 452–457. IEEE, 2004.

Amel Bouzeghoub, Kien Ngoc Do, and Claire Lecocq. A situation-based delivery of learning resources in pervasive learning. In *European Conference on Technology Enhanced Learning*, pages 450–456. Springer, 2007.

Joel C McCall and Mohan M Trivedi. Driver behavior and situation aware brake assistance for intelligent vehicles. *PROCEEDINGS-IEEE*, 95(2):374, 2007.

David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

Shardanand Upendra. *Social information filtering for music recommendation.* PhD thesis, 1994.

Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201. ACM Press/Addison-Wesley Publishing Co., 1995.

Julien Gaillard. *Systèmes de recommendation : adaptation Dynamique et Argumentation. (Recommender systems : dynamic adaptation and argumentation).* PhD thesis, University of Avignon, France, 2014.

Jon Scott Armstrong. *Principles of forecasting: a handbook for researchers and practitioners*, volume 30. Springer Science & Business Media, 2001.

Gerard Salton. Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley*, 1989.

Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.

Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

Frank Meyer. *Recommender systems in industrial contexts*. PhD thesis, University of Grenoble, 2012.

Meenakshi Sharma and Sandeep Mann. A survey of recommender systems: approaches and limitations. *International Journal of Innovations in Engineering and Technology*, 2(2):8–14, 2013.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

Romain Picot-Clémente. *Une architecture générique de Systèmes de recommandation de combinaison d'items: application au domaine du tourisme*. PhD thesis, University of Bourgogne, 2011.

Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

Amit Singhal and Gerard Salton. Automatic text browsing using vector space model. In *Proceedings of the Dual-Use Technologies and Applications Conference*, pages 318–324, 1995.

Dunja Mladenic. Machine learning used by personal webWatcher. In *Proceedings of ACAI-99 Workshop on Machine Learning and Intelligent Agents*, Chania, Crete, Greece, 1999.

Alexandros Moukas. Amalthaea information discovery and filtering using a multiagent evolving ecosystem. *Applied Artificial Intelligence*, 11(5):437–457, 1997.

Jae-wook Ahn, Peter Brusilovsky, Jonathan Grady, Daqing He, and Sue Yeon Syn. Open user profiles for adaptive news systems: help or harm? In *Proceedings of the 16th international conference on World Wide Web*, pages 11–20. ACM, 2007.

Bernardo Magnini and Carlo Strapparava. Improving user modelling with content-based techniques. In *International Conference on User Modeling*, pages 74–83. Springer, 2001.

Anna Stefani and C Strappavara. Personalizing access to web sites: The siteif project. In *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT*, volume 98, pages 20–24, 1998.

Marco Degemmis, Pasquale Lops, and Giovanni Semeraro. A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3):217–255, 2007.

Giovanni Semeraro, Pierpaolo Basile, Marco de Gemmis, and Pasquale Lops. User profiles for personalizing digital libraries. In *Handbook of Research on Digital Libraries: Design, Development, and Impact*, pages 149–158. 2009.

Stuart E Middleton, Nigel R Shadbolt, and David C De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1): 54–88, 2004.

Iván Cantador, Alejandro Bellogín, and Pablo Castells. News@ hand: A semantic web approach to recommending news. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 279–283. Springer, 2008.

Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

Upendra Shardanand and Pattie Maes. Social information filtering algorithms for automating word of mouth. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press Addison Wesley Publishing Co., 1995.

Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

Miha Grčar, Blaž Fortuna, Dunja Mladenič, and Marko Grobelnik. knn versus svm in the collaborative filtering framework. In *Data Science and Classification*, pages 251–260. Springer, 2006.

Robert Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104. ACM, 2007a.

Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proc. KDD Cup Workshop at SIGKDD'07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 39–42, 2007.

Daniel Billsus and Michael J Pazzani. User modeling for adaptive news access. *User modeling and user-adapted interaction*, 10(2-3):147–180, 2000.

Wei Chu and Seung-Taek Park. Personalized recommendation on dynamic content using predictive bilinear models. In *Proceedings of the 18th international conference on World wide web*, pages 691–700. ACM, 2009.

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

Robert Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104. ACM, 2007b.

Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR workshop on recommender systems*, volume 60. Citeseer, 1999.

Thorsten Joachims, Dayne Freitag, Tom Mitchell, et al. Webwatcher: A tour guide for the world wide web. In *IJCAI (1)*, pages 770–777. Citeseer, 1997.

Paul Cotter and Barry Smyth. Ptv: Intelligent personalised tv guides. In *AAAI/IAAI*, pages 957–964, 2000.

Ekkawut Rojsattarat and Nuanwan Soonthornphisaj. Hybrid recommendation: Combin-

ing content-based prediction and collaborative filtering. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 337–344. Springer, 2003.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document, 2000.

Yi Zhang and Jonathan Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 47–54. ACM, 2007.

Duen-Ren Liu, Pei-Yun Tsai, and Po-Huan Chiu. Personalized recommendation of popular blog articles for mobile applications. *Information Sciences*, 181(9):1552–1572, 2011.

Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134. ACM, 2002.

Paolo Massa and Bobby Bhattacharjee. Using trust in recommender systems: an experimental analysis. In *International Conference on Trust Management*, pages 221–235. Springer, 2004.

Seung-Taek Park and Wei Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 21–28. ACM, 2009.

Rana Chamsi Abu Quba. *On enhancing recommender systems by utilizing general social networks combined with users goals and contextual awareness.* PhD thesis, Lyon 1, 2015.

Jia Zhou and Tiejian Luo. A novel approach to solve the sparsity problem in collaborative filtering. In *Networking, Sensing and Control (ICNSC), 2010 International Conference on*, pages 165–170. IEEE, 2010.

Chrsistian Desrosiers and George Karypis. Solving the sparsity problem: Collaborative filtering via indirect similarities. *Department of Computer Science and Engineering University of Minnesota*, 2008.

Panagiotis Adamopoulos and Alexander Tuzhilin. On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 153–160. ACM, 2014.

Yibo Chen, Chanle Wu, Ming Xie, and Xiaojun Guo. Solving the sparsity problem in recommender systems using association retrieval. *Journal of computers*, 6(9):1896–1902, 2011.

Marijn Koolen, Toine Bogers, Antal Van Den Bosch, and Jaap Kamps. Looking for books in social media: An analysis of complex search requests. In *European Conference on Information Retrieval*, pages 184–196. Springer, 2015.

Danny Bickson. http://bickson.blogspot.fr/2012/09/harry-potter-effect-on-recommendations.html. 2012.

Saúl Vargas Sandoval. *Novelty and diversity evaluation and enhancement in recommender systems*. PhD thesis, 2015.

Òscar Celma Herrada. *Music recommendation and discovery in the long tail*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2009.

Brian McFee, Thierry Bertin-Mahieux, Daniel PW Ellis, and Gert RG Lanckriet. The million song dataset challenge. In *Proceedings of the 21st International Conference on World Wide Web*, pages 909–916. ACM, 2012.

Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

Asela Gunawardana and Guy Shani. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10(Dec):2935–2962, 2009.

Harald Steck. Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 213–220. ACM, 2013.

Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005.

Michael D Ekstrand, F Maxwell Harper, Martijn C Willemsen, and Joseph A Konstan. User perception of differences in recommender algorithms. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 161–168. ACM, 2014.

James Schaffer, Tobias Höllerer, and John O'Donovan. Hypothetical recommendation: A study of interactive profile manipulation behavior for recommender systems. In *FLAIRS Conference*, pages 507–512, 2015.

Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.

Dokyun Lee and Kartik Hosanagar. When do recommender systems work the best?: The moderating effects of product attributes and consumer reviews on recommender

performance. In *Proceedings of the 25th International Conference on World Wide Web*, pages 85–97. International World Wide Web Conferences Steering Committee, 2016.

Mojisola Helen Erdt. *Personalized Recommender Systems for Resource-based Learning-Hybrid Graph-based Recommender Systems for Folksonomies*. PhD thesis, Dissertation, Darmstadt, Technische Universität Darmstadt, 2014, 2014.

Mojisola Erdt, Florian Jomrich, Katja Schüler, Christoph Rensing, et al. Investigating crowdsourcing as an evaluation method for tel recommender systems. In *ECTEL meets ECSCW 2013: Workshop on Collaborative Technologies for Working and Learning*, volume 1047, pages 25–29, 2013.

Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. Offline and online evaluation of news recommender systems at swissinfo. ch. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 169–176. ACM, 2014.

Guy Shani, David Heckerman, and Ronen I Brafman. An mdp-based recommender system. *Journal of Machine Learning Research*, 6(Sep):1265–1295, 2005.

Amos Azaria, Avinatan Hassidim, Sarit Kraus, Adi Eshkol, Ofer Weintraub, and Irit Netanely. Movie recommender system for profit maximization. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 121–128. ACM, 2013.

Willy Shih, Stephen Kaufman, and David Spinola. *Netflix*. Harvard Business Publishing, 2009.

Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

Li Chen, Wen Wu, and Liang He. How personality influences users' needs for recom-

mendation diversity? In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 829–834. ACM, 2013.

Pinata Winoto and Tiffany Y Tang. The role of user mood in movie recommendations. *Expert Systems with Applications*, 37(8):6086–6092, 2010.

Dirk Bollen, Bart P Knijnenburg, Martijn C Willemsen, and Mark Graus. Understanding choice overload in recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 63–70. ACM, 2010.

Pearl Pu, Li Chen, and Rong Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 157–164. ACM, 2011.

Puerta Melguizo, Toine Bogers, LWJ Boves, A Deshpande, and A Bosch. What a proactive recommendation system needs: relevance, non-intrusiveness, and a new long-term memory. 2007.

Francesco Ricci. Mobile recommender systems. *Information Technology & Tourism*, 12 (3):205–231, 2010.

Reinhard Oppermann and Marcus Specht. A context-sensitive nomadic exhibition guide. In *International Symposium on Handheld and Ubiquitous Computing*, pages 127–142. Springer, 2000.

Matthias Braunhofer, Francesco Ricci, Béatrice Lamche, and Wolfgang Wörndl. A context-aware model for proactive recommender systems in the tourism domain. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pages 1070–1075. ACM, 2015.

Wen Li, Carsten Eickhoff, and Arjen P de Vries. Want a coffee?: predicting users' trails. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1171–1172. ACM, 2012b.

Daniel Gallego Vico, Wolfgang Woerndl, and Roland Bader. A study on proactive delivery of restaurant recommendations for android smartphones. In *ACM RecSys workshop on personalization in mobile applications, Chicago, USA*, 2011.

Somkiat Sae-Ueng, Sineenard Pinyapong, Akihiro Ogino, and Toshikazu Kato. Personalized shopping assistance service at ubiquitous shop space. In *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 838–843. IEEE, 2008.

Ahmed Elbery, Mustafa ElNainay, and Hesham Rakha. Proactive and reactive carpooling recommendation system based on spatiotemporal and geosocial data. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2016 IEEE 12th International Conference on*, pages 1–8. IEEE, 2016.

Daniele Quercia, Jonathan Ellis, and Licia Capra. Nurturing social networks using mobile phones. *IEEE Pervasive Computing*, 9(3):12–20, 2010.

Hong Joo Lee and Sung Joo Park. Moners: A news recommender for the mobile web. *Expert Systems with Applications*, 32(1):143–150, 2007b.

Raouia Ayachi, Imen Boukhris, Sehl Mellouli, Nahla Ben Amor, and Zied Elouedi. Proactive and reactive e-government services recommendation. *Universal Access in the Information Society*, 15(4):681–697, 2016.

Hairong Li, Steven M Edwards, and Joo-Hyun Lee. Measuring the intrusiveness of advertisements: Scale development and validation. *Journal of advertising*, 31(2):37–47, 2002.

Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gançarski. A contextual-bandit algorithm for mobile context-aware recommender system. In *International Conference on Neural Information Processing*, pages 324–331. Springer, 2012.

Ken Hinckley and Eric Horvitz. Toward more sensitive mobile phones. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 191–192. ACM, 2001.

Daniel P Siewiorek, Asim Smailagic, Junichi Furukawa, Andreas Krause, Neema Moraveji, Kathryn Reiger, Jeremy Shaffer, and Fei Lung Wong. Sensay: A context-aware mobile phone. In *ISWC*, volume 3, page 248, 2003.

Karen Kay-Lynn Liu. *A personal, mobile system for understanding stress and interruptions*. PhD thesis, Citeseer, 2004.

Gediminas Adomavicius and YoungOk Kwon. Multi-criteria recommender systems. In *Recommender systems handbook*, pages 847–880. Springer, 2015.

Tania Farinella, Sonia Bergamaschi, and Laura Po. A non-intrusive movie recommendation system. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 736–751. Springer, 2012.

K Palanivel and R Sivakumar. A study on implicit feedback in multicriteria e-commerce recommender system. *Journal of Electronic Commerce Research*, 11(2):140, 2010.

Zhongduo Lin. *Indoor Location-based Recommender System*. PhD thesis, University of Toronto, 2013.

Punam Bedi and Sumit Kr Agarwal. A situation-aware proactive recommender system. In *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*, pages 85–89. IEEE, 2012.

Nofar Dali Betzalel, Bracha Shapira, and Lior Rokach. Please, not now!: A model for timing recommendations. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 297–300. ACM, 2015.

David Milne, Paul Thomas, and Cecile Paris. Finding, weighting and describing venues: Csiro at the 2012 trec contextual suggestion track. Technical report, DTIC Document, 2012.

Bingyang Liu, Tong Wu, Xianghui Lin, Yanqin Zhong, Qian Liu, Yue Liu, and Xueqi Cheng. Ictnet at context suggestion track trec 2012. Technical report, DTIC Document, 2012.

Andrew Yates, Dave DeBoer, Hui Yang, Nazli Goharian, Steve Kunath, and Ophir Frieder. (not too) personalized learning to rank for contextual suggestion. Technical report, DTIC Document, 2012.

Maya Sappelli, Suzan Verberne, and Wessel Kraaij. Tno and run at the trec 2012 contextual suggestion track: Recommending personalized touristic sights using google places. Technical report, DTIC Document, 2012.

Marijn Koolen, Jaap Kamps, and Hugo Huurdeman. Contextual suggestion from wikitravel: Exploiting community-based suggestions. Technical report, DTIC Document, 2012.

Peilin Yang and Hui Fang. An exploration of ranking-based strategy for contextual suggestion. Technical report, DTIC Document, 2012.

Gilles Hubert and Guillaume Cabanac. Irit at trec 2012 contextual suggestion track. Technical report, DTIC Document, 2012.

Fasilkom. Fasilkom ui from universitas indonesia at trec 2012 contextual suggestion track. Technical report, DTIC Document, 2012.

Joseph L Fleiss, Bruce Levin, and Myunghee Cho Paik. *Statistical methods for rates and proportions*. John Wiley & Sons, 2013.