Identifying Context Information in Datasets

Georgia M. Kapitsaki(✉), Giouliana Kalaitzidou, Christos Mettouris, Achilleas P. Achilleos, and George A. Papadopoulos

Department of Computer Science, University of Cyprus, 1 University Avenue, Nicosia, Cyprus
{gkapi,gkalai01,mettour,achilleas,george}@cs.ucy.ac.cy

Abstract. Datasets are used in various applications assisting in performing reasoning and grouping actions on available data (e.g., clustering, classification, recommendations). Such sources of information may contain aspects relevant to context. In order to use to the fullest this context and draw useful conclusions, it is vital to have intelligent techniques that understand which portions of the dataset are relevant to context and what kind of context they represent. In this work we address the above issue by proposing a context extraction technique from existing datasets. We present a process that maps the given data of a dataset to a specific context concept. The prototype of our work is evaluated through an initial collection of datasets collected from various online sources.

Keywords: Context extraction · Dataset · Context matchmaking

1 Introduction

Context-awareness is an area that has gained tremendous interest from the research community in the latest years targeting in many cases pervasive and mobile computing systems. Adaptive, personalized services that take into account location as well as other user-related data predicate the existence of well-formed information with respect to users environment, referred to as contextual information or context. Although context is in many cases mapped to location information many definitions of context that include its different aspects apart from location, such as weather conditions, user profile information, device and netwotk connectivity, can be found in the literature [8,20]. However, the most popular definition is given by Dey and Abowd [1]: *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*. The techniques that enable the exploitation of contextual information are generally known as *context-handling* techniques, while the use of context to provide relevant information and/or services to the user, where relevancy depends on the users task, is known as *context-awareness*.

Various applications, where context is utilized for various purposes, can be found and these are mainly mobile applications and the recently emerged

H. Christiansen et al. (Eds.): CONTEXT 2015, LNAI 9405, pp. 214–225, 2015.
DOI: 10.1007/978-3-319-25591-0_16

Context-Aware Recommender Systems (CARS) that utilize context-information to provide better recommendations to end-users [4]. Context plays also a vital role in mobile context-aware applications for users on-the-move, where user surroundings and current activities are used for offering a personalized experience to users [16].

In order to be able to utilize effectively the available context information for given applications, context identification is necessary. in the framework of this work we define context identification as *the process of identifying which information constitutes context information and which not.* This process should also include more information on the kind of context addressed, i.e., one should also indicate whether specific context information refers to location data, data connected to the user, data connected to hardware devices used, etc.

In this work we address the above by introducing a process that assists in making sense of context data hidden in given datasets. Specifically, we propose a context extraction technique from an existing dataset as input source. We present a process that classifies the existing information in a given dataset as a specific context concept. The context concepts are based on a context taxonomy that we introduce for this purpose, although any context model can be used instead [6]. The prototype of our work is evaluated through an initial set of datasets containing various files that we have collected from online sources and research works, such as the one used in the multi-agent system for the care of elderly people living at home on their own [13]. This evaluation serves as proof-of-concept for the usefulness and the effectiveness of the proposed approach.

To the best of our knowledge, context extraction from datasets is a process that has not been adequately studied before. We are currently handling this task in the following way: appropriate string comparisons are performed on the dataset feature names with a context taxonomy as a reference, arguing in this manner whether a particular feature could be used as a context element by a context-aware application or not. At a second phase, a similar process is followed for the feature values, but in this case concepts with a wider sense are compared against the feature values using a lexical database that contains information on such relations between words. Although our approach is simple, we argue that it can serve as a first step towards context information extraction from datasets that can potentially enhance context-aware applications, such as Context-Aware Recommender Systems in incorporating the extracted context in their recommendation method. Identifying which information constitutes context can be a useful asset for making better use of the available data.

The rest of the paper is structured as follows. Section 2 presents the area of context modeling in recommeder systems giving at the same time a brief overview of related work on context modeling and identification. Section 3 presents our main contribution and the extraction steps proposed along with implementation details. Section 4 is dedicated to the presentation of the evaluation of our work using online datasets and to a discussion on the obtained results. Finally, Sect. 5 concludes the paper.

2 Motivation and Related Work

Recommender systems use a variety of filtering techniques and recommendation methods to provide personalized recommendations to their users. The information used is mostly retrieved from the user profile, from user's usage history, as well as from item-related information. However, these traditional recommender systems use limited or none contextual information to produce recommendations. Instead, they only focus on two dimensions: the user and the items (also called two-dimensional recommenders: *Users* and *Items* are used in order to produce *Ratings*), excluding other contextual data that could be used in the recommendation process, such as the day/time, with whom the user is with, weather conditions, etc. A typical dataset of such recommender systems includes information on the user (user ID), information on the item (item ID, item features, item price, availability, etc.) and ratings of users on items. Datasets that do not include context information are known to be two dimensional.

On the contrary, Context-Aware Recommender Systems focus on using contextual information to enhance recommendations combining *Users*, *Items* and *Context* to construct *Ratings* [3,4]. The goal is to enhance their datasets with context information so that CARS produce better, enhanced and more personalized recommendations. Context information was first utilized into the recommendation process by Adomavicius et al. by proposing three approaches: the Pre-filtering approach, the Post-filtering approach and the Multidimensional Contextual Modeling approach [3].

Based on the research work of Adomavicius and Tuzhilin [4], context can be used in two ways for producing recommendations, i) the *Recommendation via Context-Driven querying and search*, where systems use contextual information from the environment (e.g., location), the user (user profile) and the system to retrieve the most relevant items to recommend (ubiquitous and location-based systems such as systems that recommend restaurants and POIs (Points of Interest) in the user's proximity), and ii) the *Recommendation via Contextual preference elicitation and estimation* [2], where systems focus on modeling user preferences by using various methods, e.g., observing the user while interacting with a system or by receiving appropriate feedback from the user regarding the recommendations. In this paper we are dealing with the second method of incorporating the context for producing recommendations, which is used by Context-Aware Recommender Systems. CARS do not use two-dimensional datasets as with traditional recommenders; rather they face the challenge of utilizing multi-dimensional, context enriched datasets that include additional contextual dimensions besides 'users' and 'items [2]. For more information on the two ways for producing recommendations the reader may refer to previous works [2,3].

Based on the above, we argue that including and recognizing possible context elements in a given dataset to be later utilized in the recommendation process in order to produce multidimensional, context-aware recommendations by Context-Aware Recommender Systems is a useful and important process. Any Context-Aware Recommender System that uses datasets in combination with sophisticated recommendation methods such as those met in traditional recommender systems to produce recommendations should be able to produce better results in cases where the datasets used are enriched with context information. The validity of the above statement is supported by the fact that Adomavicius and Tuzhilin were the first to prove that using contextual information in CARS (from context enriched datasets) indeed enhances the recommendation process [2,3] incorporating. This observation is also validated and supported by numerous works in the CARS research literature [2,4,5,10,15,21,22].

Moreover, another class of systems to be benefited by context-enriched datasets is the Ubiquitous Context-Aware Recommender Systems class (UbiCARS) [18]. UbiCARS constitute a subset of Ubiquitous Recommender Systems [18] and utilize both ways of using context mentioned above: *Recommendation via Context-Driven querying and search* to enable the provision of recommendations on location via mobile devices (e.g., identification of near-by products), as well as *Recommendation via Contextual preference elicitation and estimation* by using context-enriched (multidimensional) datasets and context-aware recommendation techniques and methods as CARS do. UbiCARS systems can be used for in-situ products recommendations and will potentially be able to provide better recommendations than common Ubiquitous Recommender Systems, since, besides utilizing the surrounding context as Ubiquitous Recommender Systems do, they also consider the multidimensional context enriched datasets (as used by CARS) in their recommendation process.

Many other related works on context have addressed the issue of context modeling with the main motivation of using context in specific applications. Context modeling is relevant also in the framework of the current work, since it can provide the structure for representing the extracted context information [6]. Since we are focusing on CARS in this work, we are not presenting in detail context modeling techniques from other domains.

3 The Context Extraction Process

3.1 Analysis Steps

The proposed context extraction process is shown in Fig. 1. The elements of a given dataset are indicated as features with a given name and value using the terminology of machine learning. The usual case is for the first row in a separate dataset file to contain the feature names with the following rows containing the records with specific values for each feature.

The context identification process is divided into two distinct phases that operate on different level on the dataset files:

- *Phase 1 - Feature names matcher*: The feature matchmaking phase classifies a specific feature in the dataset as either a context or non-context value. This phase also specifies the context category the feature belongs to based on the introduced context taxonomy (e.g., location, user, etc.).
- *Phase 2 - Feature values matcher*: Feature values are examined in this phase. These values are also classified as context or non-context with an indication of the context category.

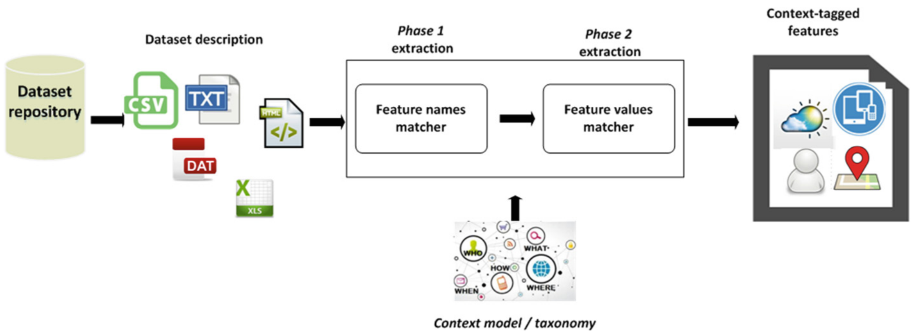

Fig. 1. The proposed context identification process.

The feature names (i.e., column names in the dataset files) correspond to the main terms whose values are contained in the dataset. These names may correspond to context-relevant elements. For the identification of such context elements and for matching purposes different string matching algorithms can be exploited. In our work the following string matching algorithms are used in the first phase of feature name matching:

- The Jaro Winkler string distance calculation algorithm [9].
- Our WordNet distance similarity algorithm introduced in a previous work [14]. Wordnet is a lexical database that retrieves similar concepts to the input word given [23]. The algorithm considers the type of connection between the examined terms. Connections of the type of same words (same), synonyms (syn), meronyms (mer), hypernyms (hyper) and related terms (rel) are considered in the following equation:

$$\sigma(n_x, n_y) = l \times \sigma_{same}(n_x, n_y) + p \times \sigma_{syn}(n_x, n_y) + q \times \sigma_{mer}(n_x, n_y)$$
$$+ r \times \sigma_{hyper}(n_x, n_y) + t \times \sigma_{rel}(n_x, n_y)$$

  The constants l, p, q, r and t express the importance of each similarity level retrieved through Wordnet. At most one of the operands in the similarity calculation will evaluate to 1.0. We have used the following values for each weight: l=1.0, p=0.7, q=0.2, r=0.2, t=0.0. Using these values 1.0 is returned only if the terms compared are exactly the same. If the terms are synonyms, then a similarity score of 0.7 is assigned.

Other algorithms that could ba considered in phase 1 can be found in the WordNet similarity algorithm from the xssm[1] (XML Schema Similarity Mapping) library in Java that returns a score between terms using a preprocessed WordNet and corpus data, and similarity based on n-grams [24].

Regarding the second phase of feature value matchmaking is performed on a different level than the first phase. Instead of using string similarity algorithms,

[1] https://code.google.com/p/xssm/.

the connection of terms in WordNet is exploited. Specifically, the hypernyms of a given term are examined for potential matching to a specific term of the context taxonomy. If an adequate hypernym is found for the feature values, then the respective feature name is considered context-relevant. For instance, if different city names (e.g., Athens, London, New York, Moscow) are indicated as feature values, these can be matched to the same hypernym synset in WordNet *town* giving an indication of a feature representing location information. Since hypernyms can reach terms in different levels, e.g., in WordNet *town* is a hypernym of *Athens* and *municipality* is a hypernym of *town*; hence, *municipality* is connected with *Athens* through 2 levels, we have selected an appropriate value for level. Based on the conducted experiments, 2 was chosen as a plausible value.

Note that this second phase considers only the first 100 entries in the dataset file. Since many datasets contain huge numbers of entries (100,000 or more), we have observed that this is an appropriate number of entries for drawing useful conclusions. Examining more rows would only add to the processing time without improving the results of the process.

Conclusions for the final characterization of a term as context-relevant or not are drawn by combining the results of the two distinct phases. Specifically, in the first phase a matching is considered succesful if the similarity algorithm returns a value higher than 0.8, whereas in the second phase this is considered for cases, where an exact hypernym of level 2 (or lower) is found. However, the significance of each phase is not the same. Phase 2 provides less accuracy, since as we observed in many dataset files only number indications are given in feature values (e.g., user or item ID numbers, year expressed in a number etc.). These cannot assist in drawing conclusions on the meaning of the values and in such cases (i.e., feature values in numeric format) phase 2 is not applied on the dataset files. Also in cases, where features values are in text format, the results of phase 1 are considered more relevant for the final results using the following weights:

$$\sigma(n_x, n_y) = 0.8 \times \sigma_{phase1}(n_x, n_y) + 0.2 \times \sigma_{phase2}(n_x, n_y)$$

Note also that important information can be found in the README files of the dataset descriptions. These are, however, not considered in the current state of our work due to the large heterogeneity of such files.

3.2 Context Model

As aforementioned a variety of context models tailored to specific domains can be found in the literature [6]. The context categorization employed in the current work has resulted from our study conducted on the state-of-the-art on context-aware systems including context-aware ubiquitous and location-aware systems, as well as context-aware recommender systems. This context categorization includes the most important context elements we have retrieved during our research. For the resulting context model captured in a taxonomy of 3 levels (main context category, subcategories, and subcategory items) in Table 1 we have used the system database from our previous work [17], as well as related literature on context models [2,4,5,10,15,21,22]. The system database of the CARS

Table 1. Context taxonomy introduced.

Context category	Context subcategory	Context terms in subcategory
User	Profile	Name, age, gender, companion, marital status, children, research interest, social role, expertise, goal, experience, employment status, education preferences, contacts, payment info
	Activity	People nearby
Environment	Weather	Temperature, humidity, Celsius, Fahrenheit, rain possibility
	Other	Season, lighting, noise level, traffic conditions
Time		Date, year, month, day, hours, minutes, seconds, timestamp, timezone
Location	Address	Street, road, city, town, municipality, prefecture, country, post code
	GPS	GPS coordinates, Latitude, longitude
System		Battery level, computing platform, bandwidth, network connectivity, communication cost, nearby resources

Context Modeling System [17] includes context models presented and used by research works in the field of context-aware recommender systems, as well as context models built by developers and experts on context-aware development at the university premises.

Please note that our aim was to build a context categorization that would be generic enough to facilitate a wide range of context-aware applications, and that our system is developed so that it can use other context categorizations as well, provided that these are given in the appropriate format.

3.3 Implementation Tools

The proposed context extraction process has been implemented in Java with the assistance of different libraries: e.g., Apache Common CSV[2] for the parsing of the CSV files and the JWNL[3] Java WordNet Library. Appropriate implemenations of the aforementioned string matching algorithms in Java were also used.

4 Evaluation and Discussion

4.1 Testing Set and Experiments

We have collected a number of datasets from various web sources. Each dataset is composed of one or more CSV files, whereas README files with more

[2] https://commons.apache.org/proper/commons-csv//.
[3] http://sourceforge.net/projects/jwordnet/.

Table 2. Datasets employed in the evaluation.

Dataset #	Application use	Number of dataset files	Source	Number of context features
1	Activity Recognition in Home Setting	4	[19]	1
2	Activity Book recommendations	3	[28]	3
3	Travel recommendations	1	[27]	5
4	Social networks: Facebook	1	*online*[a]	3
5	Social networks: Delicious	7	[7]	17
6	Social networks: last.fm	6	[7]	9
7	Social networks: MovieLens	12	[7]	24
8	Microblog spamming detection	4	*online*[b]	8
9	Wireless sensor network	4	[25]	8
10	Climate data	1	[12]	3
11	Texting zone locations	1	*online*[c]	6

[a] https://github.com/ManuelB/facebook-recommender-demo/blob/master/src/main/resources/DemoFriendsLikes.csv
[b] https://archive.ics.uci.edu/ml/datasets/microblogPCU
[c] https://catalog.data.gov/dataset/texting-zone-locations

information on the dataset use are provided in some cases. Note that, as aforementioned, although these README files may contain useful information for the dataset, they are neglected in the current prototype implementation of the context extraction tool due to their diversity and use of free text language that renders uniform processing impossible.

The datasets and their domains are depicted in Table 2. Note that the majority of files in the datasets contains headers with the feature names used. For cases, where these headers were missing, they were added by our context extractor in order to facilitate the processing phases. All datasets used are available from previous works or can be found online as indicated in the table.

4.2 Main Results and Discussion

In order to measure the results of our approach we have used the following three metrics from the information retrieval field [11,26]:

$$precision = \frac{\#correctMatchesReturned}{\#totalCorrectMatches}$$

$$recall = \frac{\#correctMatchesReturned}{\#totalMatchesReturned}$$

$$f - measure = \frac{2 \times precision \times recall}{precision + recall}$$

Some feature names and some terms in the context categories consist of more than one words. For those cases each word is examined independently for matching. If a match for any of the words is found, then the feature name or the context term respectively is considered a match as a whole. Further study of the consideration of n-grams instead of unigrams could improve the matching results.

Note that a returned match is considered correct, if the correct context category is also indicated. If only the characterizaton as context is correct, then the result is not considered correct. We have measured the above for the following cases: *Consideration only of Phase 1 results*, and *Consideration of results from both Phases*. For each of the above cases the two aforementioned string similarity algorithms were employed for the first phase (i.e., Jaro Winkler distance, our custom WordNet distance similarity algorithm).

An example of values returned for the third examined dataset of *Travel recommendations* consisting of data from TripAdvisor is shown in Table 3.The results were the same for both examined algorithms and the second phase would not add any additional information on the existing observations: using the WordNet hypernyms only *hotel city* feature values were recognized as relevant to location. This feature contained values of cities in the United States, such as Houston, Los Angeles, Oklahoma City and Boston. This is a case of good context extraction, where all relevant terms have been tagged as context-relevant but only one term has been placed in a wrong category (i.e., *user timezone* has been placed under *user* instead of *time*). Note that the terms in parenthesis in the table correspond to the context category, if it exists.

The summary of results with average precision, recall and f-measure values for all datasets in the testing set are depicted in Fig. 2. These initial results indicate that the proposed approach assists in making sense of context data that may be included in the features of the dataset files. Recall and precision values reach 0.9 in many cases. Overall phase 2 does not improve the accuracy of the results, due to the rare cases of encountering feature values in text format. For this reason, we have also selected a small weight for the similarity score of phase 2. Concerning the accuracy of phase 1 better results are observed for the Jaro

Table 3. Context identification results for the Travel recommendations dataset.

Algorithm	Terms matched as context by our approach	# of terms matched as as context correctly	Context terms in dataset	Precision	Recall
Jaro-Wrinkler (Ph1,1+2), our WordNet (Ph11+2)	user(:user), user state(:user), user timezone(:user), hotel city(:location), hotel timezone(:time)	4	user(:user), user state(:user), user timezone(:time), hotel city(:location), hotel timezone(:time)	0.8	0.8

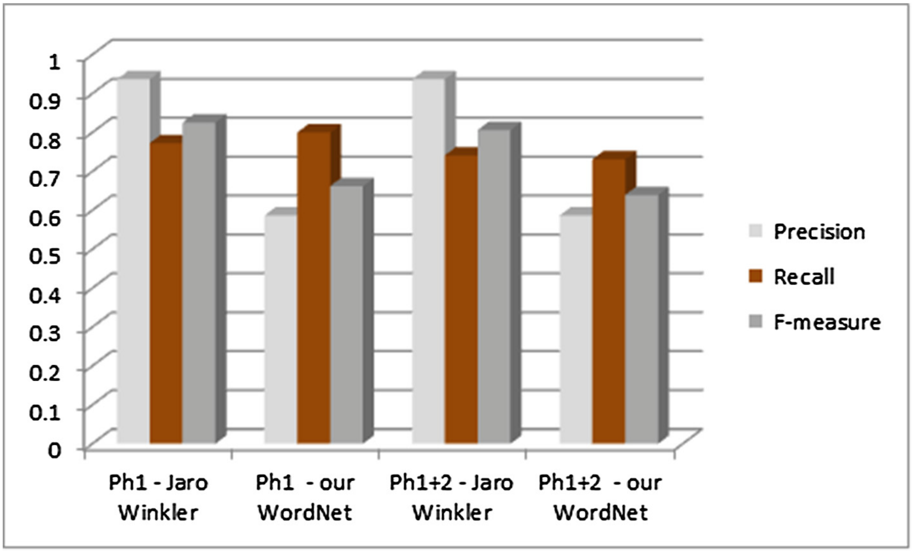

Fig. 2. Main evaluation results.

Winkler similarity algorithm that provides a higher number of terms macthed as context-relevant in comparison to the dedicated WordNet algorithm.

In general, the accuracy of the results is high, but even higher values could have been achieved. This is attributed mainly to the choice of names given to the features of datasets and also to errors in the spelling of the features names given by the dataset creators (e.g., *temperature* is spelled as *tepmrature* in the Wireless sensor network dataset). Since no common terminology exists, dataset creators are using terms that best suit their needs without conforming to any guidelines. A similar problem appears with the use of abbreviations (e.g., for state names of the United States). The utilization of the README file might assist in improving the matching results alleviating the above problems, as well as stemming or stopword removal preprocessing actions that were not employed in the framework of our work.

5 Conclusions

In this paper we have presented our work on context information identification from dataset files. We have defined a process of two phases for matching features in the datasets with context elements from a given context taxonomy. This extraction process can be a useful tool for context-aware application development and context-aware recommender systems, since it can point out context elements from huge amounts of data. We have also performed an initial evaluation of our process using a number of datasets from different application domains.

As future work we would like to utilize the results of our process for assisting software engineers in the creation of context-aware applications. We intend to focus this effort on the improvement of context-aware recommenders by suggesting the most appropriate context fields that can be used to improve the results of recommenders.

References

1. Abowd, G.D., Dey, A.K.: Towards a better understanding of context and context-awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 304–307. Springer, Heidelberg (1999)
2. Aciar, S.: Mining context information from consumers reviews. In: Proceedings of Workshop on Context-Aware Recommender System, vol. 201. ACM (2010)
3. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst. (TOIS) **23**(1), 103–145 (2005)
4. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 217–253. Springer, New York (2011)
5. Baltrunas, L., Kaminskas, M., Ricci, F., Rokach, L., Shapira, B., Luke, K.H.: Best usage context prediction for music tracks. In: Proceedings of the 2nd Workshop on Context Aware Recommender Systems (2010)
6. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive Mob. Comput. **6**(2), 161–180 (2010)
7. Cantador, I., Brusilovsky, P., Kuflik, T.: 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In: Proceedings of the 5th ACM conference on Recommender systems. RecSys 2011. ACM, New York (2011)
8. Chen, G., Kotz, D., et al.: A survey of context-aware mobile computing research. Technical Report TR2000-381, Department of Computer Science, Dartmouth College (2000)
9. Cohen, W., Ravikumar, P., Fienberg, S.: A comparison of string metrics for matching names and records. In: KDD Workshop on Data Cleaning and Object Consolidation. vol. 3, pp. 73–78 (2003)
10. Domingues, M.A., Jorge, A.M., Soares, C.: Using contextual information as virtual items on top-n recommender systems. arXiv preprint arXiv:1111.2948 (2011)
11. Marianne, H., Mathieu, L., Clémentine, N., Jean-Rémy, F.: Metamodel matching for automatic model transformation generation. In: Ober, I., Uhl, A., Völter, M., Bruel, J.-M., Czarnecki, K. (eds.) MODELS 2008. LNCS, vol. 5301, pp. 326–340. Springer, Heidelberg (2008)
12. Hansen, J., Sato, M., Ruedy, R., Lo, K., Lea, D.W., Medina-Elizade, M.: Global temperature change. Proc. Nat. Acad. Sci. **103**(39), 14288–14293 (2006)
13. Kaluža, B., Mirchevska, V., Dovgan, E., Luštrek, M., Gams, M.: An agent-based approach to care in independent living. In: de Ruyter, B., Wichert, R., Keyson, D.V., Markopoulos, P., Streitz, N., Divitini, M., Georgantas, N., Mana Gomez, A. (eds.) AmI 2010. LNCS, vol. 6439, pp. 177–186. Springer, Heidelberg (2010)
14. Kapitsaki, G.M., Achilleos, A.P.: Model matching for web services on context dependencies. In: Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services, pp. 45–53. ACM (2012)
15. Lombardi, S., Anand, S.S., Gorgoglione, M.: Context and customer behaviour in recommendation (2009)

16. Lovett, T., O'Neill, E. (eds.): Mobile Context Awareness. Springer, London (2012)
17. Mettouris, C., Papadopoulos, G.A.: Cars context modelling (2014)
18. Mettouris, C., Papadopoulos, G.A.: Ubiquitous recommender systems. Computing **96**(3), 223–257 (2014)
19. Munguia Tapia, E.: Activity recognition in the home setting using simple and ubiquitous sensors. Ph.D. thesis, Massachusetts Institute of Technology (2003)
20. Schmidt, A., Beigl, M., Gellersen, H.W.: There is more to context than location. Comput. Graph. **23**(6), 893–901 (1999)
21. Sielis, G.A., Mettouris, C., Papadopoulos, G.A., Tzanavari, A., Dols, R.M., Siebers, Q.: A context aware recommender system for creativity support tools. J. UCS **17**(12), 1743–1763 (2011)
22. Sielis, G.A., Mettouris, C., Tzanavari, A., Papadopoulos, G.A.: Context-aware recommendations using topic maps technology for the enhancement of the creativity process. In: Educational Recommender Systems and Technologies: Practices and Challenges: Practices and Challenges, p. 43 (2011)
23. Stark, M.M., Riesenfeld, R.F.: Wordnet: an electronic lexical database. In: Proceedings of 11th Eurographics Workshop on Rendering. MIT Press (1998)
24. Suen, C.Y.: n-gram statistics for natural language understanding and text processing. IEEE Trans. PAMI-Pattern Anal. Mach. Intell. **1**(2), 164–172 (1979)
25. Suthaharan, S., Alzahrani, M., Rajasegarar, S., Leckie, C., Palaniswami, M.: Labelled data collection for anomaly detection in wireless sensor networks. In: 2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 269–274. IEEE (2010)
26. Heinze, T., Voigt, K.: Metamodel matching based on planar graph edit distance. In: Gogolla, M., Tratt, L. (eds.) ICMT 2010. LNCS, vol. 6142, pp. 245–259. Springer, Heidelberg (2010)
27. Zheng, Y., Burke, R., Mobasher, B.: Differential context relaxation for context-aware travel recommendation. In: Lops, P., Huemer, C. (eds.) EC-Web 2012. LNBIP, vol. 123, pp. 88–99. Springer, Heidelberg (2012)
28. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web, pp. 22–32. ACM (2005)