Extracting context data from user reviews for recommendation: A Linked Data approach

Pedro G. Campos
Departamento de Sistemas de Información
Universidad del Bío-Bío
4081112 Concepción, Chile
pgcampos@ubiobio.cl

Nicolás Rodríguez-Artigot
Departamento de Ingeniería Informática
Universidad Autónoma de Madrid
28049 Madrid, Spain
nic.rodriguez@estudiante.uam.es

Iván Cantador
Departamento de Ingeniería Informática
Universidad Autónoma de Madrid
28049 Madrid, Spain
ivan.cantador@uam.es

ABSTRACT

In this paper we describe a novel approach to extract contextual information from user reviews, which can be exploited by context-aware recommender systems. The approach makes use of a generic, large-scale context taxonomy that is composed of semantic entities from DBpedia, the core ontology and knowledge base of the Linked Data initiative. The taxonomy is built in a semi-automatic fashion through a software tool which, on the one hand, automatically explores DBpedia by online querying for related entities and, on the other hand, allows for manual adjustments of the taxonomy. The proposed approach performs a mapping between words in the reviews and elements of the taxonomy. In this case, our tool also allows for the manual validation and correction of extracted context annotations. We describe the taxonomy creation process and the developed tool, and further present some preliminary results regarding the effectiveness of our approach.

Keywords

context modeling; context-aware recommendation; user reviews; information extraction; Linked Data; DBpedia

1. INTRODUCTION

In addition to the users' preferences –i.e., tastes and interests–, Context-Aware Recommender Systems (CARS) exploit information about the circumstances under which the users (prefer to) interact with items, such as the time of the day, the day of the week, the weather conditions, and the users' location, mood, and social companion. Studies have shown that contextual conditions may have an important, positive effect on the usefulness of recommended items [20] and, in fact, providers have reported a consistent performance improvement when context information is taken into account [2].

Despite these benefits, CARS are not used extensively. This is mainly due to the lack of available context data associated with user preferences, and the difficulty and cost to obtain it.

The simplest method to acquire context data in a recommender system consists of asking the user to explicitly state the contextual conditions as she interacts with the system items [6]. In general, however, users are not willing to provide such information because of a desire for unobtrusiveness, concerns on privacy issues, or simply the time and effort required to provide their feedback. Avoiding asking the user, another way to obtain context data is by means of physical sensors, which e.g. provide periodic records of timestamps, location coordinates, and temperature measures. Nowadays, these sensors are very common in mobile devices, but the data they generate have to be continuously processed and transformed into context representations appropriate for CARS.

An alternative technique is to identify and extract contextual information from freely given user generated contents, such as product reviews in e-commerce sites, opinion articles in web blogs, and personal posts in online social networks. Among these types of user generated contents, text reviews have been the most investigated for recommendation purposes [5][12][14].

In general, previous work on CARS has been restricted to a limited number of predefined, static context dimensions and values, assuming that context is fully observable [1]. In this paper, in contrast, we present a novel approach to extract contextual information from user reviews, under the premise that the context dimensions and values are many and unknown a priori, and may change over time, implying that context is partially observable.

Our approach makes use of a generic, large-scale context taxonomy that is composed of semantic entities –i.e., classes/categories and individuals/instances– from DBpedia [15], the Wikipedia ontology and core knowledge base of the Linked Data[1] initiative. The taxonomy is built in a semi-automatic fashion through a software tool which, on the one hand, automatically explores DBpedia by online querying for related entities and, on the other hand, allows for manual adjustments of the taxonomy (Section 2).

By means of natural language processing techniques and resources, the proposed approach performs a mapping between words in the reviews and the categories and instances in the taxonomy (Section 3). In this case, our tool also allows for the manual validation and correction of extracted context annotations. We present some preliminary results regarding the effectiveness of our approach on a well-known dataset of Amazon reviews for products in three domains, namely books, movies and music (Section 4).

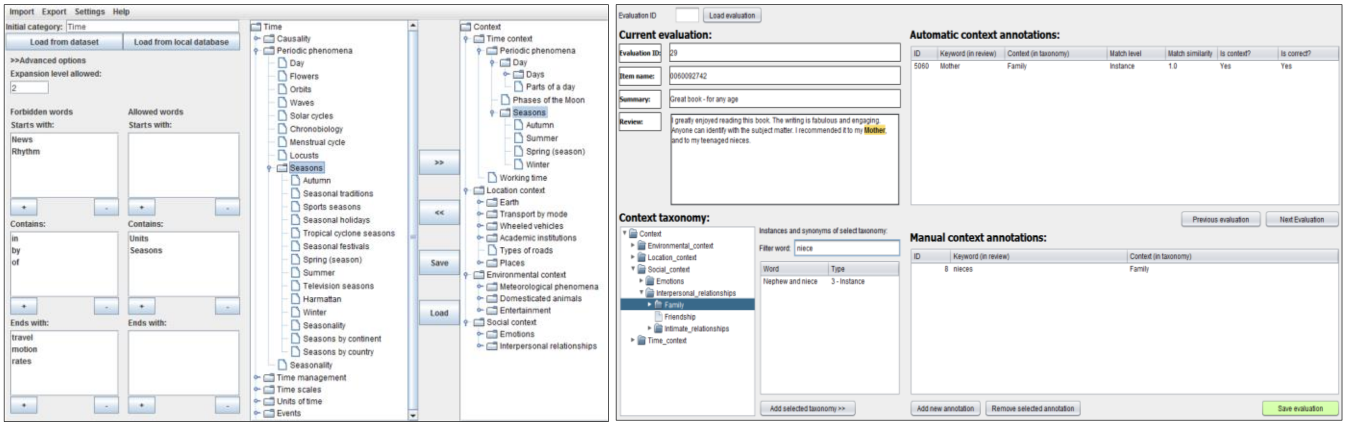
2. CONTEXT TAXONOMY

To identify context information in user reviews through the proposed approach, we first need a definition of context dimensions (a.k.a. context categories or context variables) and their respective values. As noted in [1], the context dimensions usually have a hierarchical structure, and thus can be modeled by means of a taxonomy. This is, in fact, the most common context representation followed in the literature [2].

ComplexRec'17, August 31, 2017, Como, Italy.

DOI: http://dx.doi.org/10.1145/12345.67890

[1] Linked Data, http://linkeddata.org



Initial category: Time

Expansion level allowed: 2

Starts with: News
Rhythm — Starts with:

Contains: in
by
of — Contains: Units
Seasons

Ends with: travel
motion
rates — Ends with:

Evaluation ID

Evaluation ID: 29

Item name: 0060092742

Summary: Great book - for any age

Review: I greatly enjoyed reading this book. The writing is fabulous and engaging. Anyone can identify with the subject matter. I recommended it to my Mother, and to my teenaged nieces.

ID	Keyword (in review)	Context (in taxonomy)	Match level	Match similarity	Is context?	Is correct?
5060	Mother	Family	Instance	1.0	Yes	Yes

Instances and synonyms of select taxonomy:

Filter word: niece

Word	Type
Nephew and niece	3 - Instance

ID	Keyword (in review)	Context (in taxonomy)
8	nieces	Family

Figure 1. Screenshots of the Context Taxonomy Editor (left) and the Context Annotation Validator (right) of the developed tool.

Different to existing approaches, where small and domain-dependent context are used, in this paper we advocate for the use of a generic (i.e., domain-independent), large-scale and adjustable context taxonomy. By counting with such taxonomy, it would be possible to extract and exploit context information in different domains, not being limited to recommendation purposes.

Since building from scratch, adapting and keeping updated this taxonomy manually would be highly costly, we propose to make use of collaboratively created, consensual and up-to-date repositories available on the Web. In particular, we propose to use Semantic Web-based repositories published in Linked Data [7], and more specifically, DBpedia [15], a multi-domain ontology and knowledge base created from the structured data of Wikipedia.

An important characteristic of DBpedia is that a large amount of its data is expressed using the SKOS (Simple Knowledge Organization System) vocabulary, a W3C standard model for taxonomies. By means of SKOS relations, it is possible to traverse related DBpedia categories in a hierarchical (i.e., category-subcategory) fashion. Selecting appropriate "root" categories in DBpedia (e.g., `dbc:Places`[2] for locations), and iteratively traversing subcategories –under certain restrictions–, our method automatically builds the target taxonomy, e.g., establishing that `dbc:Buildings_and_structures` and `dcb:Landforms` are subcategories of `dbc:Places`.

We surveyed previous work regarding context modeling with the aim of identifying the context dimensions considered in the literature. For each of the identified dimensions, we searched for representative DBpedia categories as root categories of the taxonomy. Next, our approach iteratively performs online queries to DBpedia for acquiring subcategories that are included in the taxonomy. To assist the process, we developed a software tool that allows browsing and modifying the taxonomy, as well as establishing the criteria that determine which (sub)categories represent context.

2.1 Building the context taxonomy

In the literature, there are several context modeling proposals, particularly in the area of pervasive and ubiquitous computing. Revising published work, we found that a major approach for context modeling is the use of taxonomies/ontologies. Table 1 summarizes some important context modeling approaches, briefly describing the context categories they consider. Based on the analyzed models, we decided to include four major context dimensions, namely *Location*, *Time*, *Environmental*, and *Social* contexts, which will have a variety of context categories. Table 2 shows the DBpedia categories selected as the root taxonomy categories for each of the above contexts.

Table 1. Main ontology-based context modeling approaches

Ref.	Context categories
[10]	**Core categories**: Person, Agent, Policy, Event, Action, Time, Space, Geo-spatial **Extended categories**: Schedule, Meeting, Contact Preference, Conditional Belief, Region, Priority
[9]	**A two-level context ontology model**, with a generic level and a domain-specific level **Basic context descriptors**: User, Resource, Location, Service, Activity, Device, Network
[8]	Instantiating the **W4 model** components: Who, What, Where, When As contextual components, *Where* is associated to the location of the fact (*What*) performed by the subject (*Who*), and *Where* refers to the time or time range associated to the fact.
[13]	Mapping between the **5W1H model** components –Who, What, Where, When, Why, How– to context items, e.g., Role, Action, Status, Location, Time, Goal

Table 2. Context dimensions and their root DBpedia categories

Context dimension	Root categories

Location context	`dbc:Places`
	`dbc:Earth`
	`dbc:Types_of_roads`
	`dbc:Academic_institutions`
	`dbc:Transport_by_mode`
	`dbc:Wheeled_vehicles`

Time context	`dbc:Periodic_phenomena` `dbc:Working_time`

Environmental context	`dbc:Meteorological_phenomena`
	`dbc:Entertainment`
	`dbc:Domesticated_animals`

Social context	`dbc:Interpersonal_relationships` `dbc:Emotions`

It is important to note that DBpedia allowed us to retrieve not only related, more specific categories by means of the `skos:broader` property, but also instances/individuals of the categories by means of the `dct:subject` property. Thus, the gathered vocabulary in the context taxonomy goes beyond the category names.

As illustrative examples, for the case of *Location context* and its root category `dbc:Wheeled_vehicles`, our approach identified the subcategories `dbc:Automobiles` and `dbc:Buses`, among

[2] `dbc:` is a prefix that stands for `http://dbpedia.org/resource/Category:`

others. For the subcategory dbc:Automobiles, it also acquired instances such as dbr:Automobile[3], dbr:Car, and dbr:Motorcar, as well as particular vehicles, e.g. dbr:Ferrari_F40 (belonging to subcategories like dbc:1990s_automobiles and dbc:Coupes).

Our tool provides a *Context Taxonomy Editor*, shown in Figure 1 (left), which browses the taxonomy, and allows the user for expanding and removing any of its categories (and corresponding subcategories and instances) by online querying DBpedia. It also allows for establishing criteria that the category names have to be satisfied to be explored or discarded by our approach. Specifically, it lets defining syntactic patterns "starts with", "contains" and "ends with" to explore/discard the categories whose names respectively start with, contain, and end with certain text; for instance, expanding those categories whose names end with the suffix *_transport*, like dbc:Road_transport and dbc:Rail_transport (which are subcategories of dbc:Transport_by_mode, the root category of the *Location context*). Moreover, the tool allows for setting the maximum depth of the taxonomy.

2.2 Enriching the context taxonomy

After navigating through the built taxonomy, we observed that there exist words describing context that were not represented as either categories or instances of the taxonomy. We realized that, in many cases, such words were synonyms of already included categories/instances. For this reason, we decided to enrich the taxonomy with such synonyms. Specifically, we obtained them from the well-known WordNet English lexical database [19].

We also detected that some morphological derivations of certain words describing context were not included in the taxonomy, but were DBpedia entities that could be retrieved via the dbo:wikiPageRedirects property, e.g., dbr:Happy redirects to dbr:Happiness. We thus further enriched the taxonomy with entities redirected by those already included in it. Table 3 shows the number of categories and instances of the final taxonomy.

Table 3. Statistics of the built taxonomy

DBpedia categories	574
Category WordNet synonyms	846
Total categories	**1420**

DBpedia instances	9591
Instance WordNet synonyms	2560
Redirected DBpedia instances	4578
Total instances	**16729**

3. CONTEXT ANNOTATION OF REVIEWS

The built taxonomy is used by our approach to identify and annotate in reviews those words that express user contextual conditions, in particular, those words that correspond to (the names of) categories or instances in the taxonomy. For such purpose, we follow two steps: first, selecting a limited number of words that could represent context and second, if possible, mapping such words to taxonomy categories.

3.1 Selecting candidate context words

In order to avoid wrong context annotations, we did not analyze all the sentences in the reviews, but only those that express personal statements and opinions of the review author, e.g., "I watched the movie with my children at home," where *children* and *home* would be annotated as social and location contexts, respectively.

To select such sentences, we made use of the Stanford CoreNLP toolkit [17]. Among other functionalities, CoreNLP generates the phrase structure tree of a sentence. Applying it to the sentences of the reviews, we processed the generated trees, and gathered the sentences where the subject is a first-person personal pronoun, *I* or *we*, the sentences where the subject is a noun phrase whose head is modified by a first-person possessive determiner, *my* or *our*, and the sentences that have a noun phrase composed by a preposition *for*, *to* or *with* followed by a first-person personal pronoun, *me* or *us*.

On each selected sentence, we first extracted candidate (simple or compound) words that may represent contextual conditions. The words used by users for describing context are nouns –e.g., evening, Saturday, restaurant, wife–, and some adjectives –e.g., sunny, cold, happy, nervous. Again, we used CoreNLP, and specifically its Part-Of-Speech tagger, to obtain the above words.

3.2 Mapping words to context categories

For every candidate word, our approach establishes whether or not the word matches certain category of the context taxonomy. In positive cases, as explained below, it generates scores that reflect the relatedness between the words and the matched categories. The resultant *(word, category, score)* tuples will be the context annotations of the input reviews. The process is done as follows.

For a fast identification of matches between review words and taxonomy categories, we took advantage of Apache Lucene[4], a well-known information retrieval software library. Hence, before performing any word-category matching, we used Lucene to create a search index for the context taxonomy. Specifically, for each taxonomy category, we created a text document containing the name of the category, the names of the instances in the category, and the synonyms of the category. We then added such document into the index. In this way, given a candidate word as input, a search in the Lucene index returns a ranked list of documents containing the word, i.e., a ranked list of categories related to the word.

In order to deal with morphological derivations –e.g. singular and plural forms of a word–, and user misspellings or deliberated word modifications –e.g. repeating the last vowel of a word as emphasis–, the words to be indexed were first lemmatized. For this reason, to effectively map a word with its corresponding context category, we further analyzed the document words retrieved by Lucene and to be compared with the queried candidate words. Specifically, for such pairs of words, (w_1, w_2), we computed the Damerau-Levenshtein distance [11][16], which measures the edit distance between two strings considering the minimum number of insertions, deletions, substitutions or transpositions of characters required to transform one string into the other. If the computed distance was lower or equal than certain threshold, then the similarity between the words was calculated as $\mathrm{sim}(w_1, w_2) = (L - 2 * D)/L$, where D is the Damerau-Levenshtein distance between w_1 and w_2, and $L = \min\{\mathrm{length}(w_1), \mathrm{length}(w_2)\}$. The category with the most similar word to the candidate was chosen as the annotated context variable, and the candidate word and its similarity were respectively taken as the context value and score of the annotation.

4. EXPERIMENTS

To preliminary assess the correctness of the context annotations generated by our approach, and their potential utility for recommendation, we conducted two experiments, namely a manual validation of annotations, and an offline evaluation of a state-of-the-art context-aware recommendation model fed with the annotations. In both experiments, we used part of the Amazon reviews dataset[5]

[3] dbr: is a prefix that stands for http://dbpedia.org/resource/

[4] Apache Lucene information retrieval software library, http://lucene.apache.org

[5] Amazon reviews dataset, http://snap.stanford.edu/data/web-Amazon.html

published in [18], whose data span a period of 18 years, including ~34.69 million text reviews (and corresponding 1-5 scale ratings) up to March 2013, provided by ~6.64 million users for ~2.44 million products. Specifically, we executed our approach to annotate the context of the first 100,000 reviews of each of the book, movie and music product review sets.

4.1 Validating the context annotations

Before evaluating the utility of the generated context annotations in recommendation, we assessed a subset of them manually. For this, we used a *Context Annotation Validator* implemented in our software tool. Shown in Figure 1 (right), the tool allows the user to load and view a review together with its metadata (id, title, summary). In a left panel with the review text, the words annotated as context are highlighted. A table on a right panel details such annotations, showing the annotated word, the associated context category, the type of match (category, instance, synonym) and the similarity score for each annotation. In the table, the user is allowed to validate whether an annotated word is context or not and, in positive cases, state whether the assigned context is correct, wrong or can be improved with a child or parent category. Moreover, the tool also allows adding manual annotations, with the possibility of browsing the taxonomy and searching for categories and instances on an interactive panel.

From the set of annotated reviews, we carefully read and manually assessed annotations in book, movie and music reviews. Table 4 shows the number of evaluated reviews and annotations, and the percentage of such annotations that were correct/context.

Table 4. Statistics and effectiveness of context annotations

	books	movies	music
Evaluated reviews	86	148	57
Evaluated annotations	100	171	86
Percentage of correct annotations	81.0%	86.6%	84.9%
Percentage of context annotations	29.0%	35.7%	45.3%
Number of manually added annotations	11	15	15

We observe that our approach obtained a relative high percentage of correct mapping cases (84.2% on average), and that most of the wrong cases were due to semantic ambiguities, which we will address in the future. The predominant contexts were *social contexts* in book reviews (e.g., daughter), *location contexts* in movie reviews (e.g., theater), and *emotional contexts* (e.g., melancholy) in music reviews.

We also notice that the number of annotations that really referred to context was low (36.7% on average). We believe that other domains, such as restaurant and hotel reviews, may have much more contextual information, and thus should be investigated.

As a lesson learnt from the conducted validation, we saw the need of investigating additional syntactic patterns with which selecting the sentences to be analyzed and annotated; for instance, we found out contextualized sentences with a third-person subject, such as *the reader/viewers…*, and *anyone interested in/looking for…*, and contextualized sentences with two-person phrases like *if you...*

4.2 Exploiting the context annotations for recommendation

In order to get initial insights about the potential benefits of exploiting our context annotations in CARS, we evaluated their effect on collaborative filtering by means of the Item Splitting (IS) context pre-filtering algorithm [3]. Hence, for each of the three considered domains, we evaluated the standard user-based (UB) and item-based (IB) k-nearest neighbor (kNN) and matrix factorization (MF) algorithms with the ratings of the reviews before and after processed by IS. Further, we also evaluated the Context Aware Matrix Factorization (CAMF) context modeling algorithm [4]. We used the implementations of those algorithms provided by the CARSKit context-aware recommendation engine [21].

From the 100,000 reviews in each domain, we selected reviews having annotations from the social, location and time context, as they showed best rate of correct context annotations. The final books, movies and music dataset used were respectively composed of 17,543, 15,983 and 14,194 reviews, 16,701, 14,340 and 12,332 users, 307, 306 and 2,540 products. Table 5 shows the achieved MAE and RMSE values for the recommendation methods in each domain using 5-fold cross-validation. Light gray indicate lower error when applying IS over the corresponding baseline. Dark gray indicate the lowest error in the corresponding column.

Table 5. Rating prediction results of collaborative filtering baselines and context-aware algorithms

	books		movies		music	
Method	MAE	RMSE	MAE	RMSE	MAE	RMSE
UB-kNN	0.8889	1.1283	0.8851	1.1590	0.7939	1.0526
IB-kNN	0.8751	1.1339	0.8566	1.1623	0.7759	1.0556
MF	0.8041	1.0759	0.8242	1.1250	0.7226	1.0179
UB-kNN + IS	0.8889	1.1284	0.8853	1.1578	0.7938	1.0524
IB-kNN + IS	0.8753	1.1337	0.8576	1.1620	0.7769	1.0561
MF + IS	0.8036	1.0743	0.8218	1.1230	0.7216	1.0148
CAMF	0.7773	1.1829	0.7635	1.2215	0.6714	1.1111

We observe that the extracted context data enable improving recommendation quality in the three domains, particularly in the case of CAMF, and also in the case of IS over MF, preliminary showing the potential of our approach.

For more argued conclusions, we should extend the experiments with other context-aware recommendation methods, and alternative evaluation metrics (e.g., ranking-based) and protocols. We also have to analyze results for each context dimension (location, time, environmental, and social) separately to conclude which of them is more/less useful in each domain.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented first a method and a software tool to semi-automatically build a context taxonomy with Linked Data. The taxonomy could be exploited by context-aware applications distinct to recommendation. Moreover, the proposed method could be used to build other types of taxonomies. For instance, it may be instantiated to describe features/aspects of items, and thus it may be exploited in feature-based opinion mining and recommendation applications.

Using the built taxonomy, we have developed a method to extract and annotate context in user reviews. We manually assessed some of the annotations, and preliminary evaluated the utility of the annotations in recommendation. We tested them with Item Splitting and Context-Aware Matrix Factorization algorithms. We are interested in evaluating additional context-aware recommendation techniques as well, and we also want to perform an exhaustive experimentation and result analysis comparing alternative approaches to index and match the context taxonomy categories in the annotation process.

6. REFERENCES

[1] Adomavicius, G., Mobasher, B., Ricci, F., and Tuzhilin, A. 2011. Context-aware recommender systems. *AI Magazine* 32(3): 67–80.

[2] Adomavicius, G., and Tuzhilin, A. 2015. Context-aware recommender systems. In *Ricci, F., Rokach, L., and Shapira, B. (Eds.), Recommender Systems Handbook – 2nd edition*, pp. 191–226.

[3] Baltrunas, L., and Ricci, F. 2009. Context-based splitting of item ratings in collaborative filtering. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, pp. 245–248.

[4] Baltrunas, L. Ludwig, B. and Ricci, F. 2011. Matrix factorization techniques for context-aware recommendation. In *Proceedings of the 5th ACM Conference on Recommender Systems*, pp. 301–304.

[5] Bauman, K., and Tuzhilin, A. 2014. Discovering contextual information from user reviews for recommendation purposes. In *Proceedings of the 1st Workshop on New Trends in Content-based Recommender Systems*, pp. 2–8.

[6] Braunhofer, M., Fernández-Tobías, I., Ricci, F. 2015. Parsimonious and adaptive contextual information acquisition in recommender systems. In *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems*, pp. 2–8.

[7] Bizer, C., Heath, T., and Berners-Lee, T. 2009. Linked Data - The story so far. *International Journal on Semantic Web and Information Systems* 5(3), 1–22.

[8] Castelli, G., Rossi, A., Mamei, M., and Zambonelli, F. 2007. A simple model and infrastructure for context-aware browsing of the world. In *Proceedings of the 5th IEEE Conference on Pervasive Computing and Communications*, 1–11.

[9] Chaari, T., Dejene, E., Laforest, F., and Scuturici, V.-M. 2007. A comprehensive approach to model and use context for adapting applications in pervasive environments. *International Journal of Systems and Software* 80(12): 1973–1992.

[10] Chen, H., Finin, T. and Joshi, A. 2005. The SOUPA ontology for pervasive computing. In *Tamma, V., Cranefield, S., Finin, T. W., and Willmott, S. (Eds.), Ontologies for Agents: Theory and Experiences*, pp. 233–258.

[11] Damerau, F. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM* 7(3): 171–176.

[12] Hariri, N., Mobasher, B., Burke, R., and Zheng, Y. 2011. Context-aware recommendation based on review mining. In *Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, pp. 30–36.

[13] Kim, J. D., Son, J., and Baik, D.K. 2012. CA5W1H onto: Ontological context-aware model based on 5W1H. *International Journal of Distributed Sensor Networks*, article ID 247346, 11.

[14] Lahlou, F. Z., Benbrahim, H., Mountassir, A., and Kassou, I. 2013. Inferring context from users' reviews for context aware recommendation. In *Proceedings of the 33th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 227–239.

[15] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellman, S., Morsey, M., van Kleef, P., Auer, S. & Bizer, C. 2015. DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6(2): 167–195.

[16] Levenshtein, V. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10(8): 707–710.

[17] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. 2014. The Stanford CoreNLP Natural Language Processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60.

[18] McAuley, J., and Leskovec, J. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 165–172.

[19] Miller, G. A. 1995. WordNet: A lexical database for English. *Communications of ACM* 38(11), 39–41.

[20] Panniello, U., Gorgoglione, M., Tuzhilin, A. 2016. In CARSs we trust: How context-aware recommendations affect customers' trust and other business performance measures of recommender systems. *Information Systems Research* 27(1), 182–196.

[21] Zheng, Y., Mobasher, B., Burke, R. 2015. CARSKit: A java-based context-aware recommendation engine. In *Proceedings of the 15th IEEE International Conference on Data Mining Workshops*, 1668–1671.