UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

EXPLORATION OF CONTEXTUAL INFORMATION EXTRACTION METHODS FOR CONSTRUCTION OF BASELINES IN THE USER REVIEW DOMAIN

CAMILA VACCARI SUNDERMANN
MARCOS AURÉLIO DOMINGUES
SOLANGE OLIVEIRA REZENDE

Nº 427

RELATÓRIOS TÉCNICOS

São Carlos – SP
Nov./2018

Instituto de Ciências Matemáticas e de Computação

Exploration of contextual information extraction methods for construction of baselines in the user review domain

CAMILA VACCARI SUNDERMANN
MARCOS AURÉLIO DOMINGUES
SOLANGE OLIVEIRA REZENDE

№

ICMC TECHNICAL REPORTS

São Carlos
November/2018

Exploration of contextual information extraction methods for construction of baselines in the user review domain

Camila Vaccari Sundermann[a]

Marcos Aurélio Domingues[b]

Solange Oliveira Rezende[a]

[a]Institute of Mathematics and Computer Science - University of São Paulo
Avenida Trabalhador São Carlense, 400, 13566-590, São Carlos, SP, Brazil
e-mail: {camilavs, solange}@icmc.usp.br

[b]Department of Informatics - State University of Maringá
Avenida Colombo, 5790, 87020-900, Maringá, PR, Brazil
e-mail: {madomingues}@uem.br

Abstract: Nowadays, with the growth of the digital universe, e-commerce and social networks, a great diversity of information, products and services is available on the Web. A recommender system can aid in user decisions like which product to buy, which movie to watch and which hotel to book. Traditional recommender systems focus on user and item data to generate recommendations. However, empirical studies indicate that context-aware approaches can produce more precise recommendations. Context-aware recommender systems are being extensively investigated. However, there is a lack of automatic methods for extracting this contextual information. With the advancement of Web 2.0 and the growing popularity of social networking and e-commerce, users have been increasingly encouraged to write reviews describing their opinions on items. There is a growing effort to incorporate into the recommender systems the important information that can be extracted from reviews. Some context extraction methods that use text mining techniques have been proposed in the literature. In this way, the objective of this work is to explore and evaluate two context extraction methods in the domain of reviews, a method based in named entities and a method based in topic hierarchies. This exploration allows the construction of baselines to be used in works that are under development in the area of context-aware recommender systems.

November/2018

Contents

1. Introduction

Nowadays, with the growth of the digital universe, e-commerce and social networks, a great diversity of information, products and services is available on the Web. Users find, while browsing, many news, products, movies and people in the social networks. With so many options, the big challenge is to identify what is really relevant and that meets the real interests and preferences of users. Thus, recommender systems have emerged with the purpose of assisting users in their choices. A recommender system is an information filtering technology that can be used to predict ratings for items (products, services, movies, among others), and/or generate a custom item ranking which may be of interest to the target user (Ricci et al., 2011). In this way, this type of system can aid in decisions like which product to buy, which movie to watch and which hotel to book.

One of the main domains that currently use recommender systems is the e-commerce domain, in which websites interact directly with customers suggesting products of interest with the aim of increasing their sales. For example, the *Amazon*[*] site, which was one of the precursors in this area, makes recommendations to users in the form: "Customers Who Viewed This Item Also Viewed..." (Linden et al., 2003). Sites from various domains such as *Netflix*[†], *Last.fm*[‡], *TripAdvisor*[§] and *Facebook*[¶] also use recommender systems. The use of such systems can represent a considerable competitive advantage on the Web.

Traditional recommender systems focus on user and item data to generate recommendations. However, empirical studies indicate that context-aware approaches can produce more precise recommendations (Adomavicius & Tuzhilin, 2005; Li et al., 2010; Hariri et al., 2011). A travel package recommender system, for example, can improve the performance of the recommendation by considering the "season of the year" context in which the user wishes to travel, since some places are most recommended in the context of "summer" while others are more recommended in the context of "winter". There are many context definitions in the literature, depending on the application area (Ricci et al., 2011). In this work, the term context is defined as any information that can be used to characterize the situation of an entity (item or user) (Dey, 2001).

Context-aware recommender systems are being extensively investigated in both the academic and corporate domains (Chen & Chen, 2015). However, some challenges are still faced by this type of system. One of the main challenges is the difficulty in acquiring

[*] `https://www.amazon.com`
[†] `https://www.netflix.com`
[‡] `http://www.last.fm`
[§] `https://www.tripadvisor.com`
[¶] `https://www.facebook.com`

contextual information to be considered when generating recommendations. There is a lack of automatic methods for extracting this type of information. Thus, effective methods and strategies are sought for this purpose and ways of identifying which contexts can be successfully extracted. On the other hand, with the advancement of Web 2.0 and the growing popularity of social networking and e-commerce, users have been increasingly encouraged to write reviews describing their opinions on items. These reviews are usually in the form of textual comments, in which users, based on their experiences, explain why they liked or disliked an item. There is a growing effort to incorporate into the recommender systems the important information that can be extracted from reviews.

As the volume of reviews is usually very large and most of it is generated in text format, it is necessary to use text mining techniques to extract contextual information. Some context extraction methods that use text mining techniques have been proposed in the literature. Domingues et al. (2014) proposed to extract named entities from the textual content of Web pages and to use such entities as contextual information in context-aware recommender systems. Sundermann et al. (2016) proposed to construct topic hierarchies of Web pages using privileged information, to extract topics from those hierarchies, and to use such topics as context in context-aware recommender systems.

In this way, the objective of this work is to explore and evaluate such context extraction methods in the domain of reviews. This exploration allows the construction of baselines to be used in works that are under development in the area of context-aware recommender systems.

2. Context-Aware Recommender Systems

According to Adomavicius & Tuzhilin (2005), recommender systems became an independent area in the mid-1990s and since then these systems are increasingly being used in various application areas. Such systems assist users by indicating which items they may be interested in, facilitating the search of such users. The items can be products, services, people, among others.

Recommender systems, known as information filtering technologies, can use various types of data to generate recommendations. In traditional systems this data is related to the items that will be suggested and the users who will receive the recommendations (Ricci et al., 2011). The traditional recommender process is known as two-dimensional, because it considers only two dimensions $User \times Item$ to generate the recommendations. However, in many applications it is also important to incorporate contextual information into the recommendation process (Adomavicius et al., 2005). For example, a travel package recommended in the summer may be different from a travel package recommended in the winter, i.e. the "season of the year" context, in this example, may interfere with user preference; a person may prefer to read politics and economy news during the week and sport and entertainment news at the weekend ("week period" context); the movie suggested for a person may depend on the context "company", i.e. who will watch with it.

Context-aware recommender systems are systems that make recommendations also considering contextual information. The importance of contextual information has been recognized by researchers and professionals in many areas (Adomavicius & Tuzhilin, 2011). Context-aware recommender systems model and predict user preferences by incorporating contextual information available in the recommender process.

Context is a concept that can have several definitions depending on the area in which it appears. The most commonly used definition was suggested by Dey (2001): "Context is any information that can be used to characterize the situation of an entity. An entity may be a person, a place, or an object that is considered relevant to the interaction between an user and an application, including the user and the application themselves".

According to Adomavicius & Tuzhilin (2011), contextual information can be applied at various stages of the recommendation process and following this criterion systems can be divided into three categories as illustrated in Figure 1: (i) **contextual pre-filtering**; (ii) **contextual modeling**; and (iii) **contextual post-filtering**.

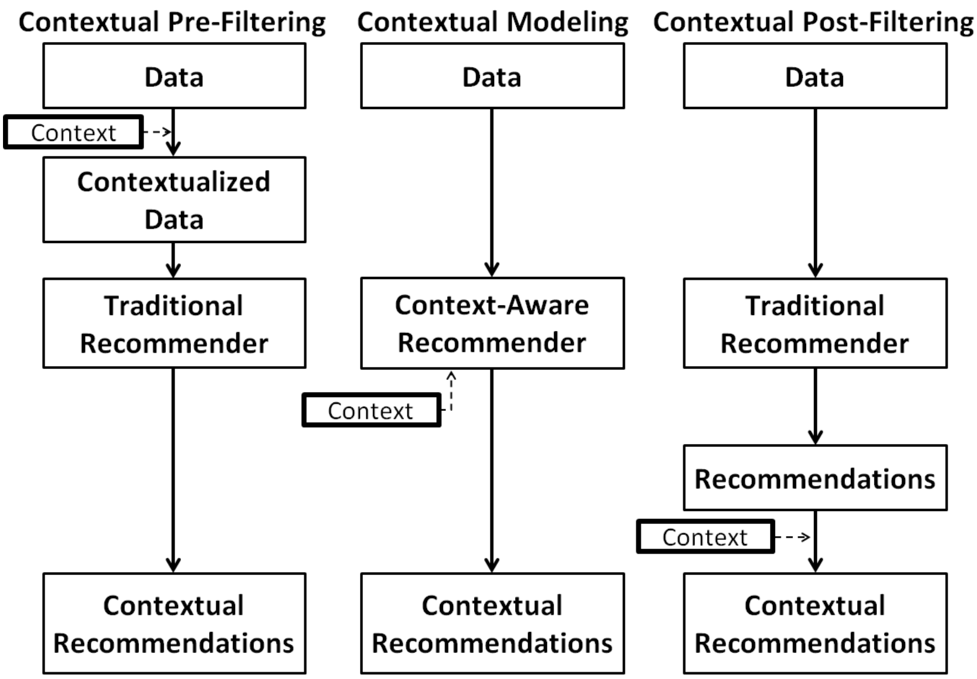

Figure 1: Classification of context-aware recommender systems (Adapted from Panniello & Gorgoglione (2012)).

In **contextual pre-filtering**, contextual information is used to select the data set that will be used for learning the recommender model. Recommendations can be made using a traditional recommender system and considering as input the selected data. An advantage of this approach is that it allows the use of any traditional recommender technique. For example, if a person wants to watch a movie on a Saturday, the context-aware movie recommender system may consider, to generate recommendations, one of the traditional techniques and, as input data, evaluations made only on Saturdays (Adomavicius & Tuzhilin, 2011).

In the **contextual post-filtering** approach, contextual information is used after the construction of a traditional recommender model to filter or reorder the recommendations, that is, the context is initially ignored. When recommendations are generated, the contextual post-filtering approach adjusts the list of recommendations obtained for each user considering contextual information. According to Adomavicius & Tuzhilin (2011), the adjustments in the list of recommendations can be made: 1) filtering the recommendations that are irrelevant in a given context; or 2) adjusting the classification of the recommendations in the list based on a certain context. For example, if a person wants to watch a movie on Sunday and it is known that on Sundays he/she only watches horror movies, then the system can only consider horror movie recommendations to display to the user.

In the **contextual modeling** approach, context is used in the recommender models, that is, the contextual information is part of the model along with the item and user data. While traditional two-dimensional functions can be used in contextual pre-filtering and post-filtering approaches, the contextual modeling approach generally uses truly multidimensional functions. These functions may represent predictive models such as decision trees, regression, probabilistic models, or others, or they may represent heuristic calculations that incorporate contextual information.

Although there are many studies in the area of context-aware recommender systems, there is a lack of automatic methods for acquiring contextual information. In the next section we present the ways in which this type of information can be extracted as well as some works in the literature that discuss the acquisition of context for context-aware recommender systems.

2.1 Extraction of Contextual Information for Context-Aware Recommender Systems

According to Adomavicius & Tuzhilin (2011), contextual information can be obtained in the form:

- **Explicit**: explicit extraction methods collect contextual information by means of directed questions, that is, the user can inform such information, for example, filling forms.
- **Implicit**: implicit extraction methods access the contextual information directly from the Web environment data. These methods do not need to interact with the user, they are able to extract information like time and location from the *log* of users' accesses.
- **Inferred**: to infer contextual information, data or text mining techniques may be used. For example, text mining techniques can be applied in user *reviews*, to extract contextual information.

In explicit methods, users are generally not interested in filling out forms if there is no motivation for doing so. In implicit methods the information obtained from access logs are usually values that can not be directly used by context-aware recommender systems. In this way, inference methods can be more effective in the search for contextual information. Data or text mining techniques can be applied to web page access and reviews to extract contextual information automatically (Lee et al., 2010).

Li et al. (2010) developed algorithms with existing natural language processing tools such as GATE* (Cunningham et al., 2002) to extract different types of contextual information from restaurant reviews.

Biancalana et al. (2013) proposed a social recommender system called Polar. This system extracts information from social networks, user reviews, and local search sites. A KEA-based extractor Jones & Paynter (2002) retrieves candidate keyphrases by using lexical methods, vector space models and Naive Bayes algorithms for learning.

Hariri et al. (2011) obtain contextual information by mining hotel reviews written by users. Their approach is based on a classifier which is trained by the description sample and their corresponding contexts.

Takehara et al. (2012) proposed a recommender system that recommends restaurants to users according to their preferences and context. The messages assessing restaurants are used as the context information which affects users in their preferences. Keywords related to the restaurants are extracted from the reviews. The influential surrounding context information is extracted from Twitter by using the keywords.

Bauman & Tuzhilin (2014) presented a method to find relevant contextual information from reviews of users. In their method, the reviews are classified as "specifics" and "generics", and the context is extracted from the specific reviews by using two methods: "*word-based*" and "*LDA-based*".

Levi et al. (2012) proposed an approach that extracts key features that are important for each context group. The weight of a feature is calculated based on its frequency in sentences appearing in reviews that belong to a specific context.

Chen & Chen (2014, 2015) extract contexts employing a keyword matching method. The authors consider that the contextual variables are "Time", "Occasion", and "Companion". Each contextual value can be assigned with different values, and each value can be defined by a set of manually-selected keywords. If any of the keywords appear in a review sentence, the sentence will be tagged with the corresponding contextual value.

In Domingues et al. (2014), topic hierarchies of Web pages were built and the topics were used as contextual information of those pages (items) in context-aware recommender systems. A non-supervised method called BC^2 (Buckshot Consensus Clustering) was used to construct the topic hierarchies.

Domingues et al. (2014) extended the work Domingues et al. (2014) also using named entities of Web pages as contextual information. Named entity recognition was performed

*http://gate.ac.uk

using *REMBRANDT* (Cardoso, 2008), a system that recognizes named entity classes, such as things, locations, organizations, people and others, in texts written in Portuguese.

In Sundermann et al. (2014), the topic hierarchies of Web pages were constructed using the LIHC method. Bag-of-words were considered as technical information and the named entities, extracted from the pages using *REMBRANDT*, as privileged information.

In Sundermann et al. (2015), the LIHC method was extended to construct topic hierarchies using two types of privileged information, besides the technical information. In this way, it was proposed to use, as contextual information, the topic hierarchies constructed using three types of information: bag-of-words, named entities (privileged information I) and domain terms (privileged information II). The domain terms were extracted using the MATE-ML method (Automatic Term Extraction based on Machine Learning) (Conrado et al., 2013; Conrado, 2014). This method uses machine learning by incorporating rich attributes of candidate terms.

Finally, in Sundermann et al. (2016), the method for context extraction using topic hierarchies constructed with the use of LIHC was presented. As privileged information was considered the terms of the domain and the named entities separately.

In this work, two context extraction methods proposed in the literature were used: the method proposed in Domingues et al. (2014), which uses named entities as contextual information; and the method proposed in Sundermann et al. (2016), which uses as context topics extracted from privileged information topic hierarchies. These methods were originally proposed and applied in the domain of Web pages. Our objective is to apply and evaluate them, observing their performance, in the domain of reviews. Both methods are detailed in the following sections.

2.2 Using named entities as contextual information in context-aware recommender systems

Named Entities are terms that represent names of people, places, and organizations. In addition, they can express time, date, money, percentage, among others. The concept of named entities is widely used in Natural Language Processing applications and, according to Sekine (2004), was born in Message Understanding Conferences (MUC).

Named entities are present in many sources of information, such as articles, web pages, blogs, reviews and social networks. In the list of most searched terms in Internet search tools, we may notice that named entities are highlighted.

The process of recognizing named entities involves identifying words or expressions belonging to named entities. First the candidate terms are identified and then a classification is made, categorizing them among the different classes of named entities. For example, be the phrase, “Carolina lives in SÃ£o Carlos and has worked as an assistant in Magazine Luiza since 2000.”, in this sentence we can identify the named entities: “Carolina”, “SÃ£o Carlos”, “Magazine Luiza” and “2000”. They can be classified as person, place, organization and date, respectively.

Some named entity classes are great examples of contextual information, such as location, time, organization, and others. In this way, they can be used as context in context-aware recommender systems. Researchers have already explored the use of named entities in context-aware recommender systems, as in the work (Domingues et al., 2014).

In this work, the method proposed in Domingues et al. (2014) (referred here as *EntityAsContext*) was applied and explored in the domain of reviews. For the recognition of named entities, the *Stanford NER tool* (Finkel et al., 2005) was used, which will be presented in more detail in Section 3.3. After the recognition of the named entities, only the terms that represent the entities are extracted, and a document collection is constructed, in which each document represents a review and is composed of all the entities present in the same.

2.3 Using topic hierarchies as contextual information in context-aware recommender systems

Topic hierarchies are clusterings of texts that are constructed with the purpose of organizing these texts automatically, allowing users to explore the collection interactively through topics that indicate the content of each group.

To perform the clustering task, we need to consider a proximity measure and a clustering strategy. The proximity measure is used to calculate the similarity between objects. In this way, similar objects are placed in the same group while being separated from dissimilar objects (Everitt et al., 2011). Clustering strategies are the methods used to form clusters. Hierarchical methods organize the textual collection into a hierarchy of groups and subgroups, which is represented by a binary tree, called *dendrogram*. In Figure 2, an example of a dendrogram is shown. The higher-level groups in the hierarchy (dendrograma) represent the most generic knowledge, while the lower-level groups represent the most specific knowledge. For the construction of topic hierarchies, after clustering, descriptors/topics are selected that indicate the content of each group and subgroup.

In the same way that texts can be organized into hierarchies, contextual information can be organized as a hierarchical structure and represented as a tree (Adomavicius &

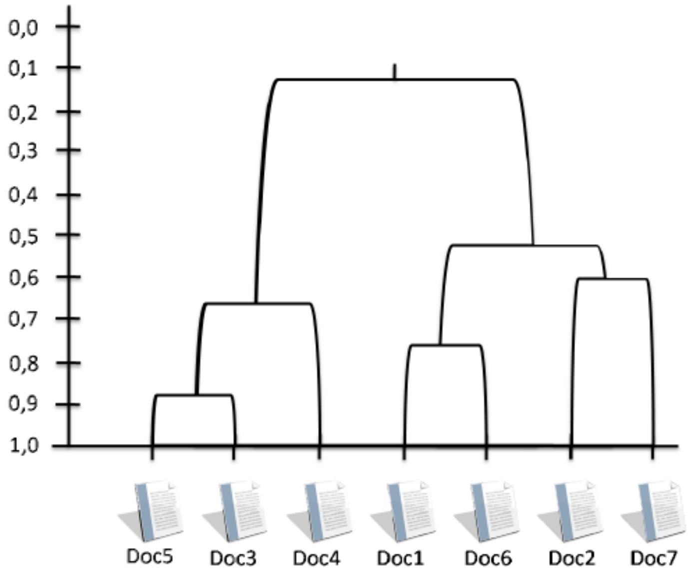

Figure 2: Example of a dendrogram that represents a topic hierarchy - (Rossi, 2011).

Tuzhilin, 2005; Panniello & Gorgoglione, 2012). Thus, Sundermann et al. (2014, 2016) proposed a method of extracting contextual information that uses topic hierarchies. Most of the literature methods for constructing topic hierarchies represent texts as a traditional bag-of-words. However, Marcacini & Rezende (2013) proposed a method called *LUPI-based Incremental Hierarchical Clustering* (LIHC) to construct topic hierarchies considering besides the technical information (bag-of-words), also richer information extracted from the texts. In Sundermann et al. (2014), Web page topic hierarchies were constructed using the LIHC method. The bag-of-words was considered as technical information and the named entities extracted by *REMBRANDT* were used as privileged information.

In Figure 3, the context extraction method proposed by Sundermann et al. (2016) (referred here as *TopicAsContext*) is illustraded. The text collection is submitted to the extraction of named entities, explained in the previous section. The documents with named entities, as well as the original textual documents, go through the pre-processing, in which stopwords are eliminated, the terms are stemmed and the representations in the vector space model are constructed. In this way, two text representations are obtained, the traditional bag-of-words (technical information) and the bag-of-entities (privileged information). Both representations are inserted in the LIHC method for the construction of the topic hierarchies. In this method, clusters of the representations are generated separately, and the combination of them is performed in the consensual grouping, in which a weight, called combination factor α, is assigned to the privileged information. After the topic hierarchies have been constructed, with the possibility of varying the value of α, the topics of the groups and subgroups are extracted. This extraction is done following the granularity configuration $\{x, y\}$, which determines that the extracted topics must have at least x documents associated with them and at maximum y documents. With this extraction strategy, we can extract more specific topics and more general topics by varying the values of x and y.

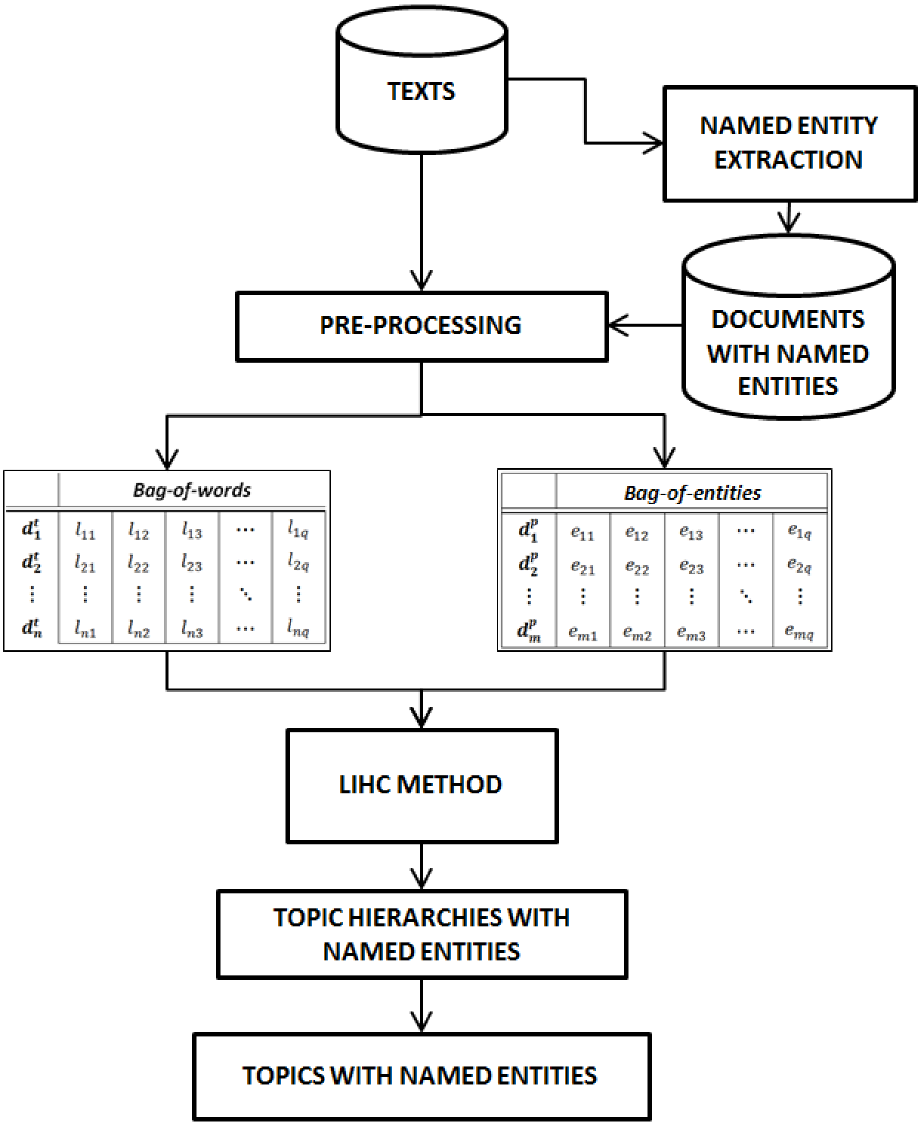

Figure 3: Overview of the method *TopicAsContext* (Sundermann et al., 2016).

In this work, the method *TopicAsContext* (Sundermann et al., 2016) was applied in the domain of reviews. As privileged information were considered the named entities extracted by the *Stanford NER tool*. Several topic hierarchies were generated by varying the classes of named entities and the value of the combination factor. In addition, different types of granularity configuration were considered for topic extraction.

3. Experimental Evaluation of the Context Extraction Methods for the Construction of Baselines

For the experimental evaluation we followed the context-aware recommender method proposed in Sundermann et al. (2018), whose overview is illustrated in Figure 4. The objective of our work was to evaluate, in the domain of reviews, two methods already proposed in the literature for context extraction. We used a review collection, which is normally composed of identifications of users and of the items evaluated by them, textual contents where the users justify their opinions about any item, date when the evaluation was made and so on. The **step 1** of the experimental evaluation is the **preprocessing**, responsible for preparing the dataset for both the recommendation and the context extraction steps. For the recommendation, the reviews are filtered, excluding those without textual content or other important information such as the user or the item identification. In addition, users, items and reviews that are less relevant to the targeted process are excluded by using the Chen & Chen (2015)'s work as reference. The exclusion criteria used consider: 1) Users with 1 review; 2) Items with less than 15 reviews; and 3) Reviews with less than 3 sentences. Besides filtering, two tasks are also performed: 1) Separation of the textual content from reviews, *i.e.* construction of a text collection in which each file represents a review textual content of the dataset; and 2) Selection of relevant data to the recommender algorithms, *i.e.* generation of a sub-dataset for the recommendation.

In the **step 2**, the text collection goes through a cleaning in order to eliminate special characters such as @, $*$, # and &. Then, the cleaned texts can be directly used by the context extraction technique (*EntityAsContext* or *TopicAsContext*) or they can pass through a normalizer in the **step 3**. Normalization aims to solve problems commonly encountered in texts written by users, like typos, spelling mistakes, abbreviations *etc.*

The main step is the **step 4**, which represents the process of extracting textual contexts from reviews by using the method proposed by Domingues et al. (2014) (*EntityAsContext*) or the method proposed by Sundermann et al. (2016) (*TopicAsContext*). In the **step 5**, the contextual dataset generated in step 4 is inserted into the **context-aware recommender systems**, along with the user and item data selected in step 1. From initial and exploratory experiments, we observed that the way in which the context is inserted in the system can influence the final result. Thus, we have used two ways of considering the context: 1) "*Context of Reviews*"; or 2) "*Context of Items*". In the first way, the contexts extracted from a review written by the user U_{ex} on the item I_{ex} are assigned to the pair U_{ex}-I_{ex}. That is, the context extracted from a review is directly

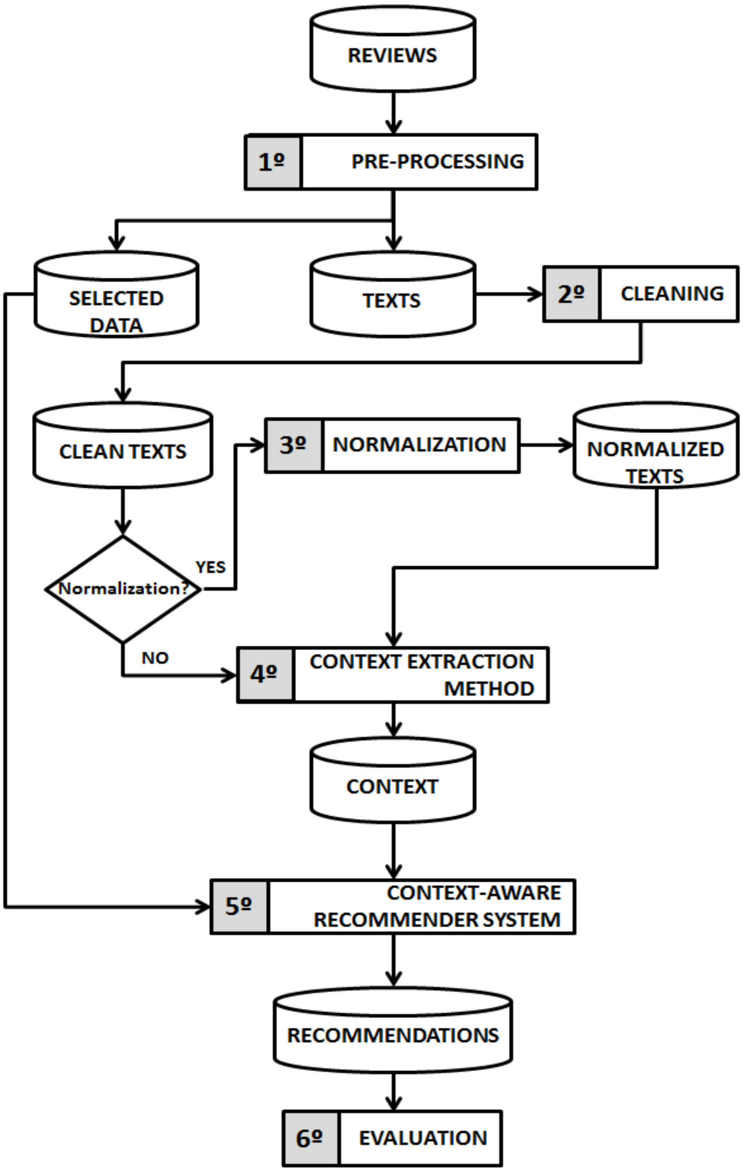

Figure 4: Overview of the context-aware recommender method proposed in Sundermann et al. (2018).

related to the user (who wrote the review) and item (which is being evaluated) pair. In the second form, the contexts extracted from all reviews written on the item I_{ex} are assigned to all the pairs $U_{I_{ex}}$-I_{ex} where $U_{I_{ex}}$ is the set of all users who evaluated the item I_{ex}. Here we are considering that the context is related to the item, no matter which user is evaluating the item. We can say that this is a way of enriching the context used in the recommendation.

The output of the **step 5** is the recommendations generated by the context-aware recommender systems using the contexts extracted by the context extraction methods (*EntityAsContext* and *TopicAsContext*). In the step 6 we evaluate the recommendations generated considering the contexts extracted by both methods.

3.1 Dataset

In the experiments we used the RecSys dataset for the recommender system challenge ACM RecSysChallenge 2013 proposed to the customization of recommendations for Yelp* users, which is a multinational company based in San Francisco, California (USA). The Yelp company has applications and a website where users can rate business establishments. In each evaluation it is possible to leave a rating, from one to five stars and also write a text explaining the opinion about the establishment.

The RecSys dataset originally consists of 11,537 items (business), 45,980 users and 229,901 reviews. However, after the step 1 of our proposal, the dataset was reduced to 2,510 items, 16,086 users and 130,632 reviews. We also got a text collection composed by 130,632 documents (textual contents of reviews) and the user-item collection for the recommendation. In the step 2, the documents were cleaned and then normalized in the step 3. In this way, two datasets were generated, *YelpClean* and *YelpCleanNormal*, both with 130,632 documents.

The datasets went through the step 4 for the extraction of contextual information by using the two context extraction methods (*EntityAsContext* and *TopicAsContext*). For the *EntityAsContext* method, the named entity classes recognized were: time, date, local, and organization. To evaluate the importance of each class as contextual information, the following combinations were considered:

1. Date
2. Date and local
3. Date, local and organization
4. Date, local and time
5. Date, organization
6. Date, organization and time
7. Date and time
8. Local
9. Local and organization
10. Local, organization and time
11. Local and time
12. Organization

*https://www.yelp.com

13. Organization and time
14. Time
15. Date, local, organization and time

Therefore, for the method *EntityAsContext*, 15 experiment variations were executed for each dataset. For the method *TopicAsContext*, we constructed 3 different topic hierarchies by varying the value of the combination factor $\alpha = 0$, $\alpha = 0.5$ and $\alpha = 1$ and considering the 15 combinations of named entities as privileged information. For the extraction of the topics, the following granularity configurations were considered: $\{2, 7\}$, $\{10, 50\}$, $\{15, 20\}$ and $\{50, 100\}$. In this way, 180 (15 named entity combinations x 3 values of α x 4 granularity configurations) combinations of experiments were executed for the method *TopicAsContext*, for each dataset.

In Tables 1 and 2, we present the number of each contextual information, the number of transactions (user x item x context) and the average of contexts by items, for *EntityAsContext* and *TopicAsContext* respectively. For the method *EntityAsContext*, we refer to each contextual information as the entity classes (*Date*, *Local*, *Org* and *Time*) linked by underline. For example, *Date_Time*, *Data_Org*, and so on. For the baseline *TopicAsContext*, the entity classes used as privileged information are linked by underline with the combination factor and the granularity configuration. For example, *DataOrg_05_15_20* represents the contextual information (topics) extracted from the topic hierarchy constructed by using the named entity classes *Data* and *Organization* as privileged information, the combination factor $\alpha = 0.5$ and the granularity configuration $\{15, 20\}$.

Table 1: Characteristics of the contextual information extracted by the method *EntityAsContext*.

Context	**YelpClean**			**YelpCleanNormal**		
	#Contexts	**#Transac.**	**#Contexts/ #Items**	**#Contexts**	**#Transac.**	**#Contexts/ #Items**
Date	2630	33411	13.6	2722	41025	16.6
Date_Local	12227	113872	45.4	7258	92674	37.0
Date_Local_Org	38003	188295	75.0	12024	113108	45.1
Date_Local_Time	13221	129443	51.6	7832	105559	42.1
Date_Org	30257	108627	43.3	8030	61739	24.6
Date_Org_Time	31242	124196	49.5	8604	74624	29.7
Date_Time	3628	48983	19.7	3296	51649	20.8
Local	9617	80431	32.1	4543	51649	20.8
Local_Org	35425	154891	61.7	9311	33603	13.6
Local_Org_Time	36431	170467	67.9	9893	84972	33.9
Local_Time	10637	96039	38.3	5125	64538	25.8
Org	27666	75219	30.0	5312	20714	8.8
Org_Time	28674	90794	36.2	5894	33603	13.6
Time	1022	15578	7.2	582	12889	6.3
Date_Local_Org_Time	38986	203866	81.2	12598	125993	50.2

Table 2: Characteristics of the contextual information extracted by the method *TopicAsContext*.

Context	YelpClean			YelpCleanNormal		
	#Contexts	#Transac.	#Contexts/ #Items	#Contexts	#Transac.	#Contexts/ #Items
Date_0_2_7	3427	12259	6.3	3356	12285	6.6
Date_0_10_50	1224	35362	15.8	1244	34853	15.3
Date_0_15_20	266	5440	4.0	253	4969	3.8
Date_0_50_100	365	40088	18.1	394	42256	18.3
Date_05_2_7	3564	13200	7.0	3627	13729	7.7
Date_05_10_50	1528	47304	22.5	1470	41543	20.0
Date_05_15_20	333	6693	4.9	315	6208	4.6
Date_05_50_100	390	40053	19.5	354	35784	17.7
Date_1_2_7	4749	16387	8.3	4520	15006	8.8
Date_1_10_50	1202	26767	12.6	981	22223	11.6
Date_1_15_20	278	5074	3.7	261	4620	3.2
Date_1_50_100	201	17752	8.7	170	14486	7.9
DateLocal_0_2_7	3545	12874	6.6	3458	12304	6.5
DateLocal_0_10_50	1365	38218	17.1	1292	34640	15.4
DateLocal_0_15_20	294	5620	4.1	290	5784	4.1
DateLocal_0_50_100	397	41420	18.5	371	41775	18.6
DateLocal_05_2_7	3637	13311	7.0	3664	13412	7.2
DateLocal_05_10_50	1593	44922	20.4	1443	40013	19.2
DateLocal_05_15_20	383	7421	4.8	321	6247	4.6
DateLocal_05_50_100	396	43121	20.6	366	38403	17.9
DateLocal_1_2_7	4304	15415	7.8	4270	14594	7.9
DateLocal_1_10_50	1346	34085	15.8	1127	28212	13.8
DateLocal_1_15_20	278	5155	3.6	269	5120	3.6
DateLocal_1_50_100	246	21088	10.3	214	21428	11.0
DateLocalOrg_0_2_7	3605	12882	6.6	3512	12554	6.6
DateLocalOrg_0_10_50	1416	37858	16.5	1356	36526	16.0
DateLocalOrg_0_15_20	283	5233	3.7	289	5543	3.9
DateLocalOrg_0_50_100	409	38830	17.1	460	46083	19.6
DateLocalOrg_05_2_7	3538	12990	6.7	3747	13847	7.4
DateLocalOrg_05_10_50	1458	40990	18.5	1595	43050	20.7
DateLocalOrg_05_15_20	311	6489	4.7	365	7155	4.9
DateLocalOrg_05_50_100	425	48289	22.3	279	31793	16.1
DateLocalOrg_1_2_7	4267	15079	7.7	4146	13855	7.5
DateLocalOrg_1_10_50	1266	31801	15.1	1079	26536	13.3
DateLocalOrg_1_15_20	277	4967	3.6	291	5430	3.9
DateLocalOrg_1_50_100	277	29874	14.4	223	22361	11.9
DateLocalTime_0_2_7	3597	12947	6.7	3462	12609	6.6
DateLocalTime_0_10_50	1340	35593	16.1	1359	36555	15.9
DateLocalTime_0_15_20	311	6007	4.1	317	6234	4.3
DateLocalTime_0_50_100	354	38468	17.6	414	41209	18.1
DateLocalTime_05_2_7	3548	13125	6.9	3766	13779	7.7
DateLocalTime_05_10_50	1580	45686	20.8	1488	41448	19.7
DateLocalTime_05_15_20	355	6967	4.9	345	6782	4.9
DateLocalTime_05_50_100	400	42164	19.4	369	40379	19.4
DateLocalTime_1_2_7	4338	15080	7.7	4120	13863	7.7
DateLocalTime_1_10_50	1309	32527	15.3	1152	28965	14.4
DateLocalTime_1_15_20	287	5305	3.7	230	4352	3.3
DateLocalTime_1_50_100	275	27996	13.5	265	27175	14.2
DateOrg_0_2_7	3510	12426	6.6	3360	12047	6.4

Table 2: (continued)

Context	YelpClean			YelpCleanNormal		
	#Contexts	#Transac.	#Contexts/ #Items	#Contexts	#Transac.	#Contexts/ #Items
DateOrg_0_10_50	1325	36577	16.4	1278	34759	15.5
DateOrg_0_15_20	321	6148	4.1	281	5452	4.0
DateOrg_0_50_100	384	39840	17.9	462	48334	21.1
DateOrg_05_2_7	3588	13376	7.1	3750	14010	7.6
DateOrg_05_10_50	1468	40770	18.6	1491	40000	18.6
DateOrg_05_15_20	333	6506	4.8	340	6524	4.5
DateOrg_05_50_100	401	47363	21.0	312	31221	15.3
DateOrg_1_2_7	4224	15145	7.9	4371	14450	7.8
DateOrg_1_10_50	1211	30053	14.0	1079	26368	12.8
DateOrg_1_15_20	260	4930	3.6	305	5592	3.7
DateOrg_1_50_100	248	25865	12.9	220	20676	10.6
DateOrgTime_0_2_7	3551	12938	6.7	3432	12424	6.6
DateOrgTime_0_10_50	1271	34412	15.3	1324	37203	16.6
DateOrgTime_0_15_20	275	5358	4.1	292	5668	4.0
DateOrgTime_0_50_100	458	49489	21.5	370	36599	16.7
DateOrgTime_05_2_7	3519	12924	6.8	3705	13596	7.3
DateOrgTime_05_10_50	1653	47846	21.2	1387	37341	17.4
DateOrgTime_05_15_20	367	7385	4.8	312	6184	4.7
DateOrgTime_05_50_100	441	47009	20.8	339	38051	18.1
DateOrgTime_1_2_7	4215	14916	7.8	4257	14533	7.8
DateOrgTime_1_10_50	1256	32739	15.3	1219	29268	14.1
DateOrgTime_1_15_20	286	5348	3.9	284	5198	3.6
DateOrgTime_1_50_100	287	29462	14.0	263	25576	12.4
DateTime_0_2_7	3360	12525	6.5	3488	13665	7.4
DateTime_0_10_50	1287	35296	15.7	1169	37615	17.2
DateTime_0_15_20	307	5811	3.9	270	6428	4.6
DateTime_0_50_100	377	41126	18.4	361	37414	17.3
DateTime_05_2_7	3468	13162	7.1	3658	12944	7.2
DateTime_05_10_50	1543	44611	21.2	1416	31494	15.1
DateTime_05_15_20	357	7277	5.1	325	5529	3.9
DateTime_05_50_100	371	38610	18.0	342	38979	18.9
DateTime_1_2_7	4586	16209	8.2	4432	14988	8.1
DateTime_1_10_50	1287	29433	13.7	1141	27012	13.2
DateTime_1_15_20	289	5386	3.5	264	4888	3.5
DateTime_1_50_100	233	21431	10.6	243	22769	11.2
Local_0_2_7	3532	12769	6.6	3661	13254	7.0
Local_0_10_50	1288	33900	15.0	1346	35831	15.8
Local_0_15_20	312	6102	4.2	293	5615	3.8
Local_0_50_100	402	41619	18.2	381	36552	16.1
Local_05_2_7	3682	13592	7.2	3741	13581	7.5
Local_05_10_50	1460	39965	18.8	1395	38338	18.7
Local_05_15_20	356	6756	4.6	339	6663	4.7
Local_05_50_100	404	47856	22.2	302	31933	16.3
Local_1_2_7	4386	15104	7.9	4341	14353	7.9
Local_1_10_50	1211	29217	14.0	1105	27118	13.5
Local_1_15_20	272	4935	3.4	296	5382	3.7
Local_1_50_100	225	23052	11.6	196	19753	10.4
LocalOrg_0_2_7	3643	13084	6.9	3484	12487	6.5
LocalOrg_0_10_50	1390	36755	16.2	1319	35832	15.5

Table 2: (continued)

Context	YelpClean			YelpCleanNormal		
	#Contexts	#Transac.	#Contexts/ #Items	#Contexts	#Transac.	#Contexts/ #Items
LocalOrg_0_15_20	312	5768	4.1	298	5650	3.9
LocalOrg_0_50_100	374	39641	18.0	441	42248	18.1
LocalOrg_05_2_7	3699	13357	7.0	3606	13041	7.0
LocalOrg_05_10_50	1449	39252	18.0	1431	40587	19.5
LocalOrg_05_15_20	345	6639	4.5	324	6413	4.8
LocalOrg_05_50_100	388	43389	20.1	318	33290	16.1
LocalOrg_1_2_7	4290	14804	7.8	4198	13805	7.5
LocalOrg_1_10_50	1256	31947	15.5	1112	29461	14.5
LocalOrg_1_15_20	288	5511	3.7	270	4928	3.5
LocalOrg_1_50_100	234	28079	14.2	247	24187	12.2
LocalOrgTime_0_2_7	3675	13226	6.9	3495	12630	6.5
LocalOrgTime_0_10_50	1333	34981	15.2	1313	34924	15.1
LocalOrgTime_0_15_20	316	5943	4.1	267	5235	3.6
LocalOrgTime_0_50_100	421	40772	18.0	402	39367	17.4
LocalOrgTime_05_2_7	3633	13205	6.9	3663	13595	7.4
LocalOrgTime_05_10_50	1418	41248	18.6	1382	37905	17.9
LocalOrgTime_05_15_20	354	7054	4.7	335	6577	4.7
LocalOrgTime_05_50_100	381	40097	1.8	333	32459	15.6
LocalOrgTime_1_2_7	4170	14508	7.6	4129	14016	7.6
LocalOrgTime_1_10_50	1349	35841	16.8	1097	27545	13.5
LocalOrgTime_1_15_20	322	6455	4.3	259	4853	3.4
LocalOrgTime_1_50_100	308	32166	15.4	259	28613	14.5
LocalTime_0_2_7	3618	13107	6.9	3248	11645	6.0
LocalTime_0_10_50	1282	33314	15.2	1102	30745	13.6
LocalTime_0_15_20	297	5553	3.9	281	5452	3.5
LocalTime_0_50_100	382	41188	18.5	245	37088	16.2
LocalTime_05_2_7	3557	13027	6.9	3507	13048	6.9
LocalTime_05_10_50	1433	42657	19.9	1195	33621	15.6
LocalTime_05_15_20	261	5371	4.1	279	5350	3.8
LocalTime_05_50_100	388	41590	19.8	338	33163	16.0
LocalTime_1_2_7	4379	15314	8.0	4478	14622	7.8
LocalTime_1_10_50	1287	31276	14.9	1099	25015	12.2
LocalTime_1_15_20	298	5675	3.9	295	5259	3.4
LocalTime_1_50_100	248	24314	12.1	228	20147	9.8
Org_0_2_7	3561	12698	8.7	3404	12291	16.5
Org_0_10_50	1290	33766	14.9	1310	35874	15.9
Org_0_15_20	275	5253	3.6	276	5335	3.7
Org_0_50_100	371	37506	19.6	357	36558	16.7
Org_05_2_7	3682	13633	7.1	3905	13899	7.6
Org_05_10_50	1408	37629	16.9	1104	29682	15.5
Org_05_15_20	313	6090	2.8	250	4942	4.1
Org_05_50_100	432	43859	20.5	241	23834	13.3
Org_1_2_7	4163	14638	7.9	4419	14542	8.0
Org_1_10_50	1307	35719	17.4	1092	27600	13.8
Org_1_15_20	303	5927	4.4	286	5221	3.8
Org_1_50_100	271	30813	15.8	215	22091	11.4
OrgTime_0_2_7	3534	12943	6.7	3348	12214	6.4
OrgTime_0_10_50	1302	33707	15.3	1262	34643	15.3
OrgTime_0_15_20	298	5589	4.0	302	5833	4.0

Table 2: (continued)

Context	YelpClean			YelpCleanNormal		
	#Contexts	#Transac.	#Contexts/ #Items	#Contexts	#Transac.	#Contexts/ #Items
OrgTime_0_50_100	376	38815	17.3	373	40465	18.4
OrgTime_05_2_7	3603	13350	7.0	3695	13386	7.2
OrgTime_05_10_50	1450	39013	17.5	1396	39118	18.1
OrgTime_05_15_20	339	6557	4.8	298	5878	4.3
OrgTime_05_50_100	361	39123	17.9	324	35376	16.7
OrgTime_1_2_7	4115	14735	7.9	4362	14497	7.8
OrgTime_1_10_50	1313	34649	16.5	1203	28928	13.8
OrgTime_1_15_20	299	5566	3.9	299	5502	3.6
OrgTime_1_50_100	263	26944	13.5	280	26438	12.5
Time_0_2_7	3374	12166	6.3	3350	12173	6.4
Time_0_10_50	1174	34392	15.4	1142	33612	15.2
Time_0_15_20	243	5055	3.8	222	4394	3.4
Time_0_50_100	392	42557	18.9	345	35161	15.7
Time_05_2_7	3411	12401	6.6	3642	13556	7.4
Time_05_10_50	1298	39393	17.6	1292	34851	17.3
Time_05_15_20	320	6718	4.5	283	5652	4.2
Time_05_50_100	439	47726	21.6	354	35655	17.8
Time_1_2_7	4693	15971	8.2	4607	15469	8.2
Time_1_10_50	1210	28142	13.3	1149	26099	12.6
Time_1_15_20	302	5551	3.5	328	5894	3.7
Time_1_50_100	209	17710	8.7	222	18509	8.7
DateLocalOrgTime_0_2_7	3677	12868	6.6	3499	12713	6.6
DateLocalOrgTime_0_10_50	1348	33795	14.9	1352	36693	16.3
DateLocalOrgTime_0_15_20	299	5623	4.1	314	6077	4.0
DateLocalOrgTime_0_50_100	389	37373	16.9	428	42921	18.8
DateLocalOrgTime_05_2_7	3621	13327	6.9	3665	13529	7.2
DateLocalOrgTime_05_10_50	1523	43991	19.7	1406	39357	18.8
DateLocalOrgTime_05_15_20	334	6735	4.7	296	6197	4.7
DateLocalOrgTime_05_50_100	389	43859	19.3	414	41257	19.8
DateLocalOrgTime_1_2_7	4297	14971	7.8	4161	14226	7.6
DateLocalOrgTime_1_10_50	1226	31672	14.9	1184	29953	14.3
DateLocalOrgTime_1_15_20	261	4895	3.4	260	4750	3.5
DateLocalOrgTime_1_50_100	286	29117	13.9	241	23028	11.6

3.2 Baseline

In this work we considered the non-contextual algorithm *Item-Based Collaborative Filtering* (IBCF) (Deshpande & Karypis, 2004) as baseline. An item-based collaborative filtering model M is a matrix representing the similarities among all pairs of items, according to a similarity measure. In this work, we used the cosine angle similarity measure defined as:

$$sim(i_1, i_2) = cos(\overrightarrow{i_1}, \overrightarrow{i_2}) = \frac{\overrightarrow{i_1} \cdot \overrightarrow{i_2}}{\| \overrightarrow{i_1} \| * \| \overrightarrow{i_2} \|}, \qquad (1)$$

where $\overrightarrow{i_1}$ and $\overrightarrow{i_2}$ are rating vectors and the operator "$\cdot$" denotes the dot-product of the two vectors. We considered binary feedback, *i.e.* the value 1 means that the user evaluated the respective item, whereas the value 0 is the opposite. Given an active user u_a and his/her set of observable items, the N recommendations are generated calculating recommendation scores for the candidate items as:

$$score(u_a, O, r) = \frac{\sum_{i \in K_r \cap O} sim(r, i)}{\sum_{i \in K_r} sim(r, i)}, \tag{2}$$

where K_r is the set of the k most similar items to the candidate item r. The N candidate items with the highest values of score are recommended to the user u_a. All the context-aware recommender algorithms used in this work were built based on the IBCF.

3.3 Supporting Tools and Methods

In the conduction of the experiments performed to evaluate the proposed context-aware method, we used TextExpansion* tool to execute the step 3 of the process, in order to normalize the text reviews to acquire better attributes. TextExpansion is based on lexicography and semantic dictionaries, and it also uses state-of-the-art techniques for semantic analysis and context detection.

The *Stanford NER*† tool (Finkel et al., 2005), better known as *CRFClassifier*, was used for the recognition of NEs. *Stanford NER* provides a general implementation of linear chain conditional random field (CRF) sequence models Lafferty et al. (2001). This NE recognizer includes a four class model trained for conference on natural language learning (CoNLL) that classifies named entities into the following classes: Local, Person, Organization and Misc. *Stanford NER* also includes a seven class model trained for MUC (Message Understanding Conferences) that recognize the classes Time, Local, Organization, Person, Money, Percent and Date; and a three class model trained on both data sets (CoNLL and MUC) for the intersection of those class sets. For this work, we used the seven class model to extract named entities from reviews.

In the topic hierarchy construction, we used the LIHC‡ tool for the hierarchical clustering of the items, which implements the LUPI-based Incremental Hierarchical Clustering method. This tool is part of Torch (Marcacini & Rezende, 2010), that is a set of tools developed to support text clustering and construction of topic hierarchies. The recommender systems used are part of the recommendation framework CARSLibrary§.

*`http://lasid.sor.ufscar.br/expansion/static/index.html`

†`https://nlp.stanford.edu/software/CRF-NER.shtml`

‡http://sites.labic.icmc.usp.br/torch/doceng2013

§`https://github.com/maddomingues/CARSlibrary`

Context-aware recommender systems (CARS) incorporate available contextual information in the recommendation process. In this work, we evaluate the effects of using the contextual information, obtained by our proposal, considering the four different context-aware recommender systems described below:

- ***C. Reduction*** (Adomavicius et al., 2005) (Pre-filtering approach): the contextual information is used as a label for filtering out those data that do not correspond to a specified context. The remaining data that passed the filter (contextualized data) is used to generate the recommendation model.

- ***DaVI-BEST*** (Domingues et al., 2013) (Contextual modeling approach): the context is used in the recommendation model, acting together with the user and item data. *DaVI-BEST* considers the contextual information as virtual items, using them along with the actual items in the recommendation model. After all contextual information are evaluated, it is selected the one which better outperforms the traditional non-contextual recommendation model.

- ***Weight PoF* and *Filter PoF*** (Panniello & Gorgoglione, 2012) (Contextual post-filtering approaches): the contextual information are used to reorder and filter out the recommendations. Firstly, the traditional algorithm is applied to build the recommendation model, ignoring the contextual information. Then, the probability of users accessing the items given the right context is calculated and multiplied by scores of items, to reorder the recommendations (*Weight PoF*) or to be used as a threshold to filter them (*Filter PoF*).

3.4 Experimental Setup and Evaluation Measures

A recommender system can be evaluated *online*, *offline* or by means of *user studies*. In this work, we executed an *offline* evaluation, based on the *All But One protocol* (Breese et al., 1998) with 10-fold cross validation, where the set of documents is partitioned into 10 subsets. For each fold, we use $n-1$ of these subsets for training and the rest for testing. The training set T_r ir used to build the recommendation model. For each user in the test set T_e, an item is hidden as a singleton set H, and the remaining items represent the set of observable items O used in the recommendation.

Based on 10-fold cross validation, we compute *Mean Average Precision* for 10 recommendations ($MAP@10$) and, to compare two recommendation algorithms, we applied the two-sided paired t-test with a 95% confidence level. To generate the top 10 recommendations with IBCF algorithm, we considered the 4 most similar items. For the *Filter PoF* algorithm, we used the value 0.1 as a threshold to filter out the recommendations.

3.5 Results

The results are grouped into two sets, "*Context of Reviews*" and "*Context of Items*", because we carried out the experiments considering the context in two ways, as mentioned previously. As already mentioned, the *DaVI-BEST* algorithm evaluates all contextual information and selects the one which better outperforms the traditional recommender model (i.e. IBCF - without context). If the use of the contextual information does not present better results, the recommendations are generated by IBCF algorithm, that was what happened in almost all the cases of our experiments. Therefore, in this paper, the *DaVI-BEST* results are not discussed. For the baseline IBCF, that does not use context, we obtained as result a $MAP@10$ value equals to 0.0215.

The results are presented into four subsections, representing the two methods and the two ways of considering the context. In this way, in Subsection 3.5.1, we present the results of the *EntityAsContext* method considering the "*Context of Reviews*". The results of the same method, but considering the "*Context of Items*", are presented in Subsection 3.5.2. Finally, the results of the method "TopicAsContext", considering the "*Context of Reviews*" and the "*Context of Items*", are presented in Subsections 3.5.3 and 3.5.4, respectively.

3.5.1 EntityAsContext considering the "Context of Reviews"

In Table 3, we present the values of $MAP@10$ for the method *EntityAsContext* considering the *"Context of Reviews"*. We note that for the *YelpClean* dataset, the best results were presented using the *Local_Org* context. That is, the combination of the "Local" and "Organization" classes generated more precise recommendations for the *C. Reduction* and *Weight PoF* algorithms. Already for the algorithm *Filter PoF*, the best result was presented using the *Org* context ("Organization" class).

For the *YelpCleanNormal* dataset, the algorithms *C. Reduction* and *Weight PoF* presented the best results using the context *Date_Local_Org* (combination of the "Date", "Local" and "Organization" classes). While the *Filter PoF* algorithm presented better performance considering the contexts *Org* and *Org_Time*.

Table 3: Comparing the context-aware recommendation algorithms using contexts of the method *EntityAsContext* against the non-contextual baseline IBCF. The values that are statistically different than IBCF (p-value>0.05) are together with a asterisk and the values that are better than IBCF are in boldface (considering *"Context of Reviews"*).

Context	IBCF	YelpClean			YelpCleanNormal		
		C. Reduc.	***Weight PoF***	***Filter PoF***	***C. Reduc.***	***Weight PoF***	***Filter PoF***
Date	0.0215	0.0041*	0.0042*	0.0013*	0.0037*	0.0037*	0.0013*
Date_Local	0.0215	0.0115*	0.0118*	0.0013*	0.0108*	0.0110*	0.0012*
Date_Local_Org	0.0215	0.0141*	0.0145*	0.0018*	0.0134*	0.0135*	0.0014*
Date_Local_Time	0.0215	0.0118*	0.0119*	0.0013*	0.0109*	0.0111*	0.0012*
Date_Org	0.0215	0.0115*	0.0120*	0.0036*	0.0072*	0.0075*	0.0026*
Date_Org_Time	0.0215	0.0112*	0.0115*	0.0033*	0.0076*	0.0078*	0.0014*
Date_Time	0.0215	0.0025*	0.0026*	0.0020*	0.0027*	0.0029*	0.0016*
Local	0.0215	0.0095*	0.0096*	0.0018*	0.0052*	0.0055*	0.0016*
Local_Org	0.0215	0.0143*	0.0147*	0.0023*	0.0048*	0.0051*	0.0029*
Local_Org_Time	0.0215	0.0135*	0.0140*	0.0021*	0.0094*	0.0095*	0.0016*
Local_Time	0.0215	0.0100*	0.0104*	0.0012*	0.0070*	0.0071*	0.0010*
Org	0.0215	0.0098*	0.0113*	0.0068*	0.0027*	0.0031*	0.0026*
Org_Time	0.0215	0.0102*	0.0111*	0.0054*	0.0048*	0.0051*	0.0029*
Time	0.0215	0.0032*	0.0032*	0.0026*	0.0025*	0.0029*	0.0012*
Date_Local_Org_Time	0.0215	0.0141*	0.0143*	0.0016*	0.0136*	0.0138*	0.0019*

In general, the results of the context-aware algorithms using named entities such as "Context of Reviews" were lower than the results of the IBCF algorithm with statistical significance. Thus, the contexts extracted by this method did not improve the performance of the recommendation for this dataset. In addition, normalization impaired the accuracy of the recommendations, since the results for the dataset *YelpCleanNormal* were lower.

By analyzing the characteristics of the extracted contexts, Table 1, we noted that the largest number of transactions/context/context per item may have given the best results for the *C. Reduction* and *Weight PoF* algorithms. For the algorithm *Filter PoF*, no pattern associated with the characteristics presented in Table 1 was noted.

3.5.2 EntityAsContext considering the "Context of Items"

Since normalization did not result in better values of $MAP@10$ for *"Context of Reviews"*, we considered only the *YelpClean* dataset for the *"Context of Items"* experiments. In Table 4, we present the results for such dataset. We analyzed that the *"Context of Items"* greatly improves the performance of the recommendation. All these results were better than the previous results, when considering the *"Context of Reviews"*. We can say that the *"Context of Items"* enriches the contextual information used by the system.

The algorithms *C. Reduction* and *Weight PoF* presented better results than the IBCF with statistical significance. The best results for the 3 context-aware algorithms were presented considering the context *Time*.

Table 4: Comparing the context-aware recommendation algorithms using contexts of the method *EntityAsContext* against the non-contextual baseline IBCF. The values that are statistically different than IBCF (p-value>0.05) are together with a asterisk and the values that are better than IBCF are in boldface (considering *"Context of Items"*).

Context	IBCF	YelpClean		
		C. Reduc.	***Weight PoF***	***Filter PoF***
Date	0.0215	**0.0386***	**0.0400***	0.0047*
Date_Local	0.0215	**0.0352***	**0.0369***	0.0053*
Date_Local_Org	0.0215	**0.0354***	**0.0367***	0.0057*
Date_Local_Time	0.0215	**0.0354***	**0.0369***	0.0010*
Date_Org	0.0215	**0.0348***	**0.0362***	0.0052*
Date_Org_Time	0.0215	**0.0346***	**0.0362***	0.0053*
Date_Time	0.0215	**0.0379***	**0.0389***	0.0061*
Local	0.0215	**0.0357***	**0.0374***	0.0054*
Local_Org	0.0215	**0.0353***	**0.0366***	0.0059*
Local_Org_Time	0.0215	**0.0349***	**0.0363***	0.0057*
Local_Time	0.0215	**0.0350***	**0.0365***	0.0053*
Org	0.0215	**0.0350***	**0.0368***	0.0055*
Org_Time	0.0215	**0.0348***	**0.0363***	0.0051*
Time	0.0215	**0.0394***	**0.0409***	0.0077*
Date_Local_Org_Time	0.0215	**0.0352***	**0.0367***	0.0055*

In Figure 5, the results of the method *EntityAsContext* are displayed in three graphs. The first graph shows the results for the dataset *YelpClean* using the *"Context of Reviews"*. The third graph shows the results for the dataset *YelpCleanNoraml* using the *"Context of Reviews"*. And finally, in the third we look at the results for the dataset *YelpClean* using the *"Context of Items"*. By analyzing the graphs, we note what we have discussed previously: the results for the dataset *YelpClean* were slightly higher than the results for the dataset *YelpCleanNormal* and the results using the *"Context of Items"* were higher than the results when the *"Context of Reviews"* was used.

3.5.3 TopicAsContext considering the "Context of Reviews"

The results of the method *TopicAsContext*, Table 5, were generally lower than the results of the method *EntityAsContext* and, therefore, lower than the IBCF with statistical significance. The text normalization improved the results in a few cases, as observed in Table 6.

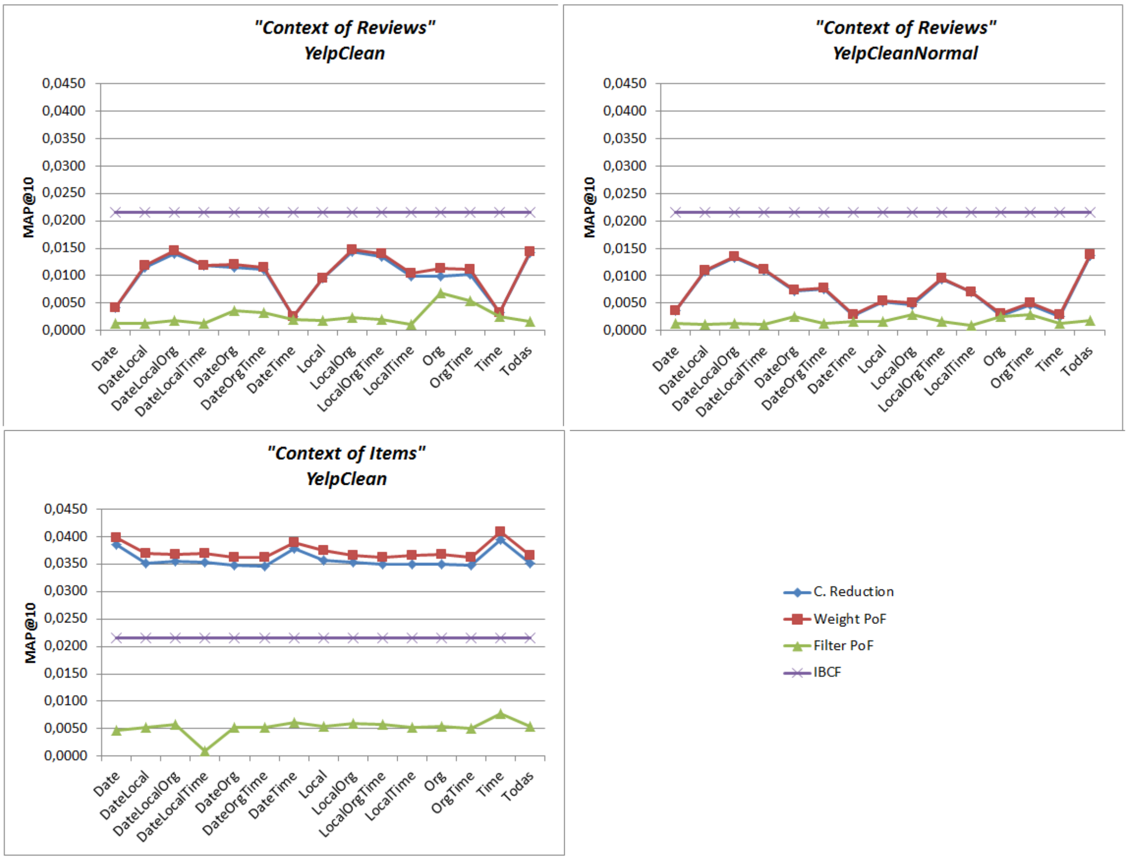

Figure 5: The graphics with the results of the method *EntityAsContext*.

Table 5: Comparing the context-aware recommendation algorithms using contexts of the method *TopicAsContext* against the non-contextual baseline IBCF. The values that are statistically different than IBCF (p-value>0.05) are together with a asterisk and the values that are better than IBCF are in boldface (considering *"Context of Reviews"*).

Context	**IBCF**	**YelpClean**			**YelpCleanNormal**		
		C. Reduc.	***Weight PoF***	***Filter PoF***	***C. Reduc.***	***Weight PoF***	***Filter PoF***
Date_0_2_7	0.0215	0.0021∗	0.0026∗	0.0029∗	0.0013∗	0.0013∗	0.0016∗
Date_0_10_50	0.0215	0.0026∗	0.0027∗	0.0033∗	0.0026∗	0.0027∗	0.0016∗
Date_0_15_20	0.0215	0.0029∗	0.0029∗	0.0002∗	0.0070∗	0.0070∗	0.0046∗
Date_0_50_100	0.0215	0.0022∗	0.0023∗	0.0019∗	0.0012∗	0.0012∗	0.0022∗
Date_05_2_7	0.0215	0.0028∗	0.0028∗	0.0003∗	0.0064∗	0.0064∗	0.0026∗
Date_05_10_50	0.0215	0.0008∗	0.0010∗	0.0014∗	0.0015∗	0.0015∗	0.0002∗
Date_05_15_20	0.0215	0.0021∗	0.0021∗	0.0007∗	0.0024∗	0.0024∗	0.0032∗
Date_05_50_100	0.0215	0.0015∗	0.0015∗	0.0004∗	0.0017∗	0.0017∗	0.0027∗
Date_1_2_7	0.0215	0.0020∗	0.0020∗	0.0013∗	0.0016∗	0.0016∗	0.0011∗
Date_1_10_50	0.0215	0.0023∗	0.0024∗	0.0014∗	0.0021∗	0.0018∗	0.0033∗
Date_1_15_20	0.0215	0.0023∗	0.0023∗	0.0010∗	0.0013∗	0.0013∗	0.0016∗
Date_1_50_100	0.0215	0.0019∗	0.0020∗	0.0015∗	0.0039∗	0.0039∗	0.0041∗
DateLocal_0_2_7	0.0215	0.0031∗	0.0031∗	0.0029∗	0.0026∗	0.0026∗	0.0004∗
DateLocal_0_10_50	0.0215	0.0016∗	0.0016∗	0.0014∗	0.0038∗	0.0037∗	0.0035∗
DateLocal_0_15_20	0.0215	0.0016∗	0.0016∗	0.0014∗	0∗	0∗	0∗
DateLocal_0_50_100	0.0215	0.0012∗	0.0013∗	0.0031∗	0.0034∗	0.0035∗	0.0041∗

Table 5: (continued)

Context	IBCF	YelpClean			YelpCleanNormal		
		C. Reduc.	***Weight PoF***	***Filter PoF***	***C. Reduc.***	***Weight PoF***	***Filter PoF***
DateLocal_05_2_7	0.0215	0.0024∗	0.0024∗	0.0014∗	0.0017∗	0.0018∗	0.0018∗
DateLocal_05_10_50	0.0215	0.0016∗	0.0016∗	0.0015∗	0.0019∗	0.0020∗	0.0022∗
DateLocal_05_15_20	0.0215	0.0017∗	0.0017∗	0.0036∗	0.0024∗	0.0024∗	0.0032∗
DateLocal_05_50_100	0.0215	0.0004∗	0.0004∗	0.0003∗	0.0017∗	0.0017∗	0.0027∗
DateLocal_1_2_7	0.0215	0.0020∗	0.0020∗	0.0032∗	0.0010∗	0.0010∗	0.0008∗
DateLocal_1_10_50	0.0215	0.0017∗	0.0018∗	0.0003∗	0.0018∗	0.0018∗	0.0010∗
DateLocal_1_15_20	0.0215	0.0086∗	0.0086∗	0.0028∗	0.0033∗	0.0037∗	0.0045∗
DateLocal_1_50_100	0.0215	0.0013∗	0.0013∗	0.0026∗	0.0029∗	0.0029∗	0.0055∗
DateLocalOrg_0_2_7	0.0215	0.0044∗	0.0044∗	0.0055∗	0.0036∗	0.0036∗	0.0019∗
DateLocalOrg_0_10_50	0.0215	0.0021∗	0.0023∗	0.0021∗	0.0023∗	0.0023∗	0.0020∗
DateLocalOrg_0_15_20	0.0215	0.0022∗	0.0020∗	0.0033∗	0.0040∗	0.0040∗	0.0051∗
DateLocalOrg_0_50_100	0.0215	0.0013∗	0.0012∗	0.0011∗	0.0009∗	0.0009∗	0.0008∗
DateLocalOrg_05_2_7	0.0215	0.0020∗	0.0020∗	0.0017∗	0.0010∗	0.0012∗	0.0037∗
DateLocalOrg_05_10_50	0.0215	0.0042∗	0.0043∗	0.0029∗	0.0021∗	0.0022∗	0.0010∗
DateLocalOrg_05_15_20	0.0215	0.0041∗	0.0058∗	0.0035∗	0.0047∗	0.0047∗	0.0025∗
DateLocalOrg_05_50_100	0.0215	0.0016∗	0.0016∗	0.0011∗	0.0019∗	0.0018∗	0.0020∗
DateLocalOrg_1_2_7	0.0215	0.0014∗	0.0014∗	0.0017∗	0.0034∗	0.0034∗	0.0056∗
DateLocalOrg_1_10_50	0.0215	0.0023∗	0.0026∗	0.0020∗	0.0038∗	0.0038∗	0.0037∗
DateLocalOrg_1_15_20	0.0215	0.0042∗	0.0042∗	0.0081∗	0.0038∗	0.0038∗	0.0043∗
DateLocalOrg_1_50_100	0.0215	0.0035∗	0.0035∗	0.0010∗	0.0018∗	0.0018∗	0.0019∗
DateLocalTime_0_2_7	0.0215	0.0021∗	0.0021∗	0.0021∗	0.0050∗	0.0051∗	0.0037∗
DateLocalTime_0_10_50	0.0215	0.0023∗	0.0025∗	0.0016∗	0.0019∗	0.0022∗	0.0007∗
DateLocalTime_0_15_20	0.0215	0.0034∗	0.0034∗	0.0033∗	0.0021∗	0.0021∗	0.0042∗
DateLocalTime_0_50_100	0.0215	0.0026∗	0.0027∗	0.0037∗	0.0029∗	0.0029∗	0.0009∗
DateLocalTime_05_2_7	0.0215	0.0034∗	0.0034∗	0.0044∗	0.0034∗	0.0036∗	0.0013∗
DateLocalTime_05_10_50	0.0215	0.0020∗	0.0021∗	0.0020∗	0.0026∗	0.0026∗	0.0014∗
DateLocalTime_05_15_20	0.0215	0.0029∗	0.0029∗	0.0013∗	0.0018∗	0.0018∗	0.0009∗
DateLocalTime_05_50_100	0.0215	0.0016∗	0.0014∗	0.0020∗	0.0016∗	0.0017∗	0.0013∗
DateLocalTime_1_2_7	0.0215	0.0019∗	0.0019∗	0.0035∗	0.0035∗	0.0035∗	0.0029∗
DateLocalTime_1_10_50	0.0215	0.0019∗	0.0019∗	0.0010∗	0.0027∗	0.0027∗	0.0032∗
DateLocalTime_1_15_20	0.0215	0.0016∗	0.0016∗	0.0018∗	0.0022∗	0.0022∗	0.0034∗
DateLocalTime_1_50_100	0.0215	0.0030∗	0.0030∗	0.0024∗	0.0037∗	0.0036∗	0.0015∗
DateOrg_0_2_7	0.0215	0.0011∗	0.0012∗	0.0017∗	0.0038∗	0.0038∗	0.0064∗
DateOrg_0_10_50	0.0215	0.0015∗	0.0020∗	0.0017∗	0.0005∗	0.0013∗	0.0012∗
DateOrg_0_15_20	0.0215	0.0041∗	0.0041∗	0.0043∗	0.0041∗	0.0051∗	0.0019∗
DateOrg_0_50_100	0.0215	0.0023∗	0.0023∗	0.0035∗	0.0065∗	0.0066∗	0.0077∗
DateOrg_05_2_7	0.0215	0.0012∗	0.0012∗	0.0009∗	0.0024∗	0.0024∗	0.0019∗
DateOrg_05_10_50	0.0215	0.0032∗	0.0032∗	0.0028∗	0.0035∗	0.0037∗	0.0015∗
DateOrg_05_15_20	0.0215	0.0024∗	0.0025∗	0.0055∗	0.0044∗	0.0045∗	0.0037∗
DateOrg_05_50_100	0.0215	0.0018∗	0.0019∗	0.0017∗	0.0040∗	0.0043∗	0.0093∗
DateOrg_1_2_7	0.0215	0.0033∗	0.0033∗	0.0016∗	0.0065∗	0.0066∗	0.0077∗
DateOrg_1_10_50	0.0215	0.0016∗	0.0022∗	0.0025∗	0.0023∗	0.0023∗	0.0039∗
DateOrg_1_15_20	0.0215	0.0018∗	0.0029∗	0.0025∗	0.0017∗	0.0017∗	0.0006∗
DateOrg_1_50_100	0.0215	0.0010∗	0.0010∗	0.0021∗	0.0027∗	0.0027∗	0.0020∗
DateOrgTime_0_2_7	0.0215	0.0016∗	0.0016∗	0.0007∗	0.0019∗	0.0019∗	0.0025∗
DateOrgTime_0_10_50	0.0215	0.0015∗	0.0016∗	0.0020∗	0.0042∗	0.0042∗	0.0044∗
DateOrgTime_0_15_20	0.0215	0.0024∗	0.0026∗	0.0051∗	0.0026∗	0.0026∗	0.0027∗
DateOrgTime_0_50_100	0.0215	0.0019∗	0.0019∗	0.0018∗	0.0015∗	0.0015∗	0.0010∗
DateOrgTime_05_2_7	0.0215	0.0030∗	0.0030∗	0.0026∗	0.0036∗	0.0036∗	0.0010∗

Table 5: (continued)

Context	IBCF	YelpClean			YelpCleanNormal		
		C. Reduc.	***Weight PoF***	***Filter PoF***	***C. Reduc.***	***Weight PoF***	***Filter PoF***
DateOrgTime_05_10_50	0.0215	0.0012∗	0.0015∗	0.0017∗	0.0011∗	0.0011∗	0.0005∗
DateOrgTime_05_15_20	0.0215	0.0019∗	0.0019∗	0.0012∗	0.0003∗	0.0003∗	0.0026∗
DateOrgTime_05_50_100	0.0215	0.0029∗	0.0029∗	0.0025∗	0.0018∗	0.0019∗	0.0019∗
DateOrgTime_1_2_7	0.0215	0.0008∗	0.0008∗	0.0016∗	0.0024∗	0.0025∗	0.0014∗
DateOrgTime_1_10_50	0.0215	0.0026∗	0.0026∗	0.0043∗	0.0021∗	0.0021∗	0.0010∗
DateOrgTime_1_15_20	0.0215	0.0051∗	0.0053∗	0.0078∗	0.0021∗	0.0021∗	0.0045∗
DateOrgTime_1_50_100	0.0215	0.0014∗	0.0014∗	0.0020∗	0.0044∗	0.0044∗	0.0012∗
DateTime_0_2_7	0.0215	0.0020∗	0.0020∗	0.0004∗	0.0017∗	0.0018∗	0.0012∗
DateTime_0_10_50	0.0215	0.0012∗	0.0013∗	0.0014∗	0.0019∗	0.0019∗	0.0019∗
DateTime_0_15_20	0.0215	0.0041∗	0.0041∗	0.0021∗	0.0015∗	0.0015∗	0.0040∗
DateTime_0_50_100	0.0215	0.0017∗	0.0017∗	0.0011∗	0.0022∗	0.0022∗	0.0033∗
DateTime_05_2_7	0.0215	0.0022∗	0.0022∗	0.0008∗	0.0011∗	0.0011∗	0.0023∗
DateTime_05_10_50	0.0215	0.0019∗	0.0019∗	0.0018∗	0.0032∗	0.0032∗	0.0023∗
DateTime_05_15_20	0.0215	0.0008∗	0.0008∗	0.0020∗	0.0070∗	0.0070∗	0.0073∗
DateTime_05_50_100	0.0215	0.0039∗	0.0039∗	0.0044∗	0.0022∗	0.0023∗	0.0025∗
DateTime_1_2_7	0.0215	0.0030∗	0.0030∗	0.0019∗	0.0006∗	0.0006∗	0.0004∗
DateTime_1_10_50	0.0215	0.0018∗	0.0018∗	0.0012∗	0.0036∗	0.0039∗	0.0010∗
DateTime_1_15_20	0.0215	0.0061∗	0.0060∗	0.0028∗	0.0065∗	0.0065∗	0.0036∗
DateTime_1_50_100	0.0215	0.0008∗	0.0008∗	0.0009∗	0.0009∗	0.0009∗	0.0010∗
Local_0_2_7	0.0215	0.0020∗	0.0020∗	0.0023∗	0.0021∗	0.0021∗	0.0026∗
Local_0_10_50	0.0215	0.0023∗	0.0024∗	0.0024∗	0.0032∗	0.0032∗	0.0025∗
Local_0_15_20	0.0215	0.0033∗	0.0033∗	0.0050∗	0.0031∗	0.0031∗	0.0005∗
Local_0_50_100	0.0215	0.0024∗	0.0025∗	0.0017∗	0.0023∗	0.0023∗	0.0007∗
Local_05_2_7	0.0215	0.0012∗	0.0015∗	0.0011∗	0.0013∗	0.0013∗	0.0018∗
Local_05_10_50	0.0215	0.0014∗	0.0019∗	0.0017∗	0.0025∗	0.0025∗	0.0018∗
Local_05_15_20	0.0215	0.0031∗	0.0031∗	0.0017∗	0.0010∗	0.0010∗	0.0016∗
Local_05_50_100	0.0215	0.0031∗	0.0033∗	0.0046∗	0.0009∗	0.0009∗	0.0013∗
Local_1_2_7	0.0215	0.0011∗	0.0011∗	0.0012∗	0.0008∗	0.0008∗	0.0010∗
Local_1_10_50	0.0215	0.0021∗	0.0021∗	0.0058∗	0.0008∗	0.0008∗	0.0023∗
Local_1_15_20	0.0215	0.0008∗	0.0008∗	0.0015∗	0.0013∗	0.0013∗	0.0019∗
Local_1_50_100	0.0215	0.0018∗	0.0018∗	0.0012∗	0.0023∗	0.0023∗	0.0018∗
LocalOrg_0_2_7	0.0215	0.0028∗	0.0029∗	0.0014∗	0.0028∗	0.0033∗	0.0029∗
LocalOrg_0_10_50	0.0215	0.0028∗	0.0031∗	0.0019∗	0.0029∗	0.0029∗	0.0015∗
LocalOrg_0_15_20	0.0215	0.0020∗	0.0020∗	0.0034∗	0.0059∗	0.0075∗	0.0061∗
LocalOrg_0_50_100	0.0215	0.0032∗	0.0034∗	0.0019∗	0.0037∗	0.0038∗	0.0021∗
LocalOrg_05_2_7	0.0215	0.0018∗	0.0018∗	0.0020∗	0.0013∗	0.0013∗	0.0018∗
LocalOrg_05_10_50	0.0215	0.0018∗	0.0019∗	0.0016∗	0.0030∗	0.0031∗	0.0029∗
LocalOrg_05_15_20	0.0215	0.0027∗	0.0030∗	0.0028∗	0.0024∗	0.0024∗	0.0032∗
LocalOrg_05_50_100	0.0215	0.0018∗	0.0018∗	0.0006∗	0.0012∗	0.0013∗	0.0011∗
LocalOrg_1_2_7	0.0215	0.0025∗	0.0023∗	0.0038∗	0.0030∗	0.0030∗	0.0033∗
LocalOrg_1_10_50	0.0215	0.0033∗	0.0033∗	0.0006∗	0.0016∗	0.0016∗	0.0011∗
LocalOrg_1_15_20	0.0215	0.0057∗	0.0057∗	0.0023∗	0.0017∗	0.0017∗	0.0006∗
LocalOrg_1_50_100	0.0215	0.0017∗	0.0021∗	0.0011∗	0.0014∗	0.0018∗	0.0018∗
LocalOrgTime_0_2_7	0.0215	0.0046∗	0.0048∗	0.0048∗	0.0009∗	0.0009∗	0.0017∗
LocalOrgTime_0_10_50	0.0215	0.0020∗	0.0020∗	0.0029∗	0.0041∗	0.0043∗	0.0033∗
LocalOrgTime_0_15_20	0.0215	0.0045∗	0.0056∗	0.0022∗	0.0017∗	0.0017∗	0.0014∗
LocalOrgTime_0_50_100	0.0215	0.0033∗	0.0034∗	0.0033∗	0.0023∗	0.0026∗	0.0013∗
LocalOrgTime_05_2_7	0.0215	0.0019∗	0.0019∗	0.0018∗	0.0029∗	0.0029∗	0.0021∗
LocalOrgTime_05_10_50	0.0215	0.0018∗	0.0018∗	0.0021∗	0.0013∗	0.0013∗	0.0014∗

Table 5: (continued)

Context	IBCF	YelpClean			YelpCleanNormal		
		C. Reduc.	***Weight PoF***	***Filter PoF***	***C. Reduc.***	***Weight PoF***	***Filter PoF***
LocalOrgTime_05_15_20	0.0215	0.0018∗	0.0028∗	0.0011∗	0.0058∗	0.0058∗	0.0035∗
LocalOrgTime_05_50_100	0.0215	0.0021∗	0.0021∗	0.0019∗	0.0019∗	0.0021∗	0.0015∗
LocalOrgTime_1_2_7	0.0215	0.0005∗	0.0005∗	0.0012∗	0.0008∗	0.0008∗	0.0012∗
LocalOrgTime_1_10_50	0.0215	0.0013∗	0.0013∗	0.0018∗	0.0007∗	0.0007∗	0.0011∗
LocalOrgTime_1_15_20	0.0215	0.0011∗	0.0011∗	0.0026∗	0.0015∗	0.0015∗	0.0037∗
LocalOrgTime_1_50_100	0.0215	0.0014∗	0.0014∗	0.0026∗	0.0023∗	0.0023∗	0.0019∗
LocalTime_0_2_7	0.0215	0.0051∗	0.0051∗	0.0096∗	0.0052∗	0.0052∗	0.0031∗
LocalTime_0_10_50	0.0215	0.0030∗	0.0030∗	0.0024∗	0.0015∗	0.0015∗	0.0009∗
LocalTime_0_15_20	0.0215	0.0019∗	0.0019∗	0.0037∗	0.0003∗	0.0003∗	0.0003∗
LocalTime_0_50_100	0.0215	0.0014∗	0.0015∗	0.0016∗	0.0018∗	0.0018∗	0.0023∗
LocalTime_05_2_7	0.0215	0.0020∗	0.0020∗	0.0023∗	0.0036∗	0.0036∗	0.0018∗
LocalTime_05_10_50	0.0215	0.0015∗	0.0016∗	0.0024∗	0.0010∗	0.0013∗	0.0010∗
LocalTime_05_15_20	0.0215	0.0023∗	0.0023∗	0.0012∗	0.0003∗	0.0003∗	0.0003∗
LocalTime_05_50_100	0.0215	0.0033∗	0.0036∗	0.0040∗	0.0023∗	0.0020∗	0.0031∗
LocalTime_1_2_7	0.0215	0.0014∗	0.0014∗	0.0007∗	0.0014∗	0.0020∗	0.0021∗
LocalTime_1_10_50	0.0215	0.0012∗	0.0012∗	0.0021∗	0.0015∗	0.0015∗	0.0021∗
LocalTime_1_15_20	0.0215	0.0036∗	0.0036∗	0.0060∗	0.0048∗	0.0048∗	0.0053∗
LocalTime_1_50_100	0.0215	0.0015∗	0.0015∗	0.0028∗	0.0021∗	0.0021∗	0.0021∗
Org_0_2_7	0.0215	0.0036∗	0.0037∗	0.0032∗	0.0036∗	0.0040∗	0.0027∗
Org_0_10_50	0.0215	0.0033∗	0.0031∗	0.0021∗	0.0024∗	0.0026∗	0.0037∗
Org_0_15_20	0.0215	0.0029∗	0.0029∗	0.0039∗	0.0046∗	0.0046∗	0.0014∗
Org_0_50_100	0.0215	0.0029∗	0.0029∗	0.0046∗	0.0013∗	0.0013∗	0.0024∗
Org_05_2_7	0.0215	0.0023∗	0.0024∗	0.0028∗	0.0038∗	0.0039∗	0.0055∗
Org_05_10_50	0.0215	0.0014∗	0.0014∗	0.0029∗	0.0011∗	0.0011∗	0.0034∗
Org_05_15_20	0.0215	0.0017∗	0.0017∗	0.0014∗	0.0048∗	0.0049∗	0.0016∗
Org_05_50_100	0.0215	0.0053∗	0.0057∗	0.0059∗	0.0012∗	0.0012∗	0.0018∗
Org_1_2_7	0.0215	0.0045∗	0.0047∗	0.0027∗	0.0029∗	0.0029∗	0.0039∗
Org_1_10_50	0.0215	0.0028∗	0.0028∗	0.0024∗	0.0028∗	0.0028∗	0.0030∗
Org_1_15_20	0.0215	0.0038∗	0.0038∗	0.0033∗	0.0043∗	0.0043∗	0.0055∗
Org_1_50_100	0.0215	0.0022∗	0.0028∗	0.0017∗	0.0020∗	0.0020∗	0.0015∗
OrgTime_0_2_7	0.0215	0.0017∗	0.0017∗	0.0013∗	0.0035∗	0.0035∗	0.0022∗
OrgTime_0_10_50	0.0215	0.0022∗	0.0023∗	0.0006∗	0.0031∗	0.0031∗	0.0028∗
OrgTime_0_15_20	0.0215	0.0047∗	0.0047∗	0.0039∗	0.0034∗	0.0034∗	0.0017∗
OrgTime_0_50_100	0.0215	0.0026∗	0.0027∗	0.0024∗	0.0015∗	0.0015∗	0.0008∗
OrgTime_05_2_7	0.0215	0.0016∗	0.0015∗	0.0021∗	0.0016∗	0.0016∗	0.0023∗
OrgTime_05_10_50	0.0215	0.0027∗	0.0029∗	0.0037∗	0.0038∗	0.0042∗	0.0039∗
OrgTime_05_15_20	0.0215	0.0074∗	0.0087∗	0.0080∗	0.0009∗	0.0009∗	0.0017∗
OrgTime_05_50_100	0.0215	0.0032∗	0.0032∗	0.0025∗	0.0026∗	0.0026∗	0.0015∗
OrgTime_1_2_7	0.0215	0.0009∗	0.0011∗	0.0024∗	0.0042∗	0.0042∗	0.0036∗
OrgTime_1_10_50	0.0215	0.0023∗	0.0024∗	0.0012∗	0.0045∗	0.0045∗	0.0023∗
OrgTime_1_15_20	0.0215	0.0031∗	0.0031∗	0.0020∗	0.0044∗	0.0056∗	0.0060∗
OrgTime_1_50_100	0.0215	0.0018∗	0.0020∗	0.0027∗	0.0007∗	0.0008∗	0.0005∗
Time_0_2_7	0.0215	0.0037∗	0.0040∗	0.0020∗	0.0011∗	0.0011∗	0.0014∗
Time_0_10_50	0.0215	0.0012∗	0.0012∗	0.0012∗	0.0023∗	0.0023∗	0.0019∗
Time_0_15_20	0.0215	0.0045∗	0.0045∗	0.0003∗	0.0003∗	0.0003∗	0.0003∗
Time_0_50_100	0.0215	0.0022∗	0.0025∗	0.0024∗	0.0038∗	0.0039∗	0.0025∗
Time_05_2_7	0.0215	0.0004∗	0.0004∗	0.0016∗	0.0005∗	0.0005∗	0.0019∗
Time_05_10_50	0.0215	0.0008∗	0.0008∗	0.0004∗	0.0009∗	0.0009∗	0.0012∗
Time_05_15_20	0.0215	0.0016∗	0.0016∗	0.0026∗	0.0029∗	0.0029∗	0.0029∗

Table 5: (continued)

Context	IBCF	YelpClean			YelpCleanNormal		
		C. Reduc.	***Weight PoF***	***Filter PoF***	***C. Reduc.***	***Weight PoF***	***Filter PoF***
Time_05_50_100	0.0215	0.0014*	0.0014*	0.0015*	0.0004*	0.0004*	0.0023*
Time_1_2_7	0.0215	0.0015*	0.0015*	0.0024*	0.0007*	0.0007*	0.0007*
Time_1_10_50	0.0215	0.0034*	0.0034*	0.0020*	0.0024*	0.0024*	0.0019*
Time_1_15_20	0.0215	0.0009*	0.0009*	0.0010*	0.0011*	0.0012*	0.0008*
Time_1_50_100	0.0215	0.0011*	0.0011*	0.0018*	0.0006*	0.0006*	0.0017*
DateLocalOrgTime_0_2_7	0.0215	0.0050*	0.0050*	0.0078*	0.0037*	0.0037*	0.0018*
DateLocalOrgTime_0_10_50	0.0215	0.0023*	0.0026*	0.0019*	0.0047*	0.0047*	0.0059*
DateLocalOrgTime_0_15_20	0.0215	0.0004*	0.0004*	0.0004*	0.0021*	0.0021*	0.0027*
DateLocalOrgTime_0_50_100	0.0215	0.0025*	0.0026*	0.0033*	0.0018*	0.0019*	0.0022*
DateLocalOrgTime_05_2_7	0.0215	0.0025*	0.0029*	0.0028*	0.0020*	0.0021*	0.0033*
DateLocalOrgTime_05_10_50	0.0215	0.0038*	0.0039*	0.0041*	0.0022*	0.0022*	0.0030*
DateLocalOrgTime_05_15_20	0.0215	0.0044*	0.0044*	0.0052*	0.0035*	0.0047*	0.0067*
DateLocalOrgTime_05_50_100	0.0215	0.0030*	0.0030*	0.0014*	0.0007*	0.0007*	0.0025*
DateLocalOrgTime_1_2_7	0.0215	0.0034*	0.0034*	0.0021*	0.0013*	0.0013*	0.0014*
DateLocalOrgTime_1_10_50	0.0215	0.0051*	0.0051*	0.0021*	0.0016*	0.0016*	0.0004*
DateLocalOrgTime_1_15_20	0.0215	0.0039*	0.0039*	0.0019*	0.0072*	0.0072*	0.0095*
DateLocalOrgTime_1_50_100	0.0215	0.0009*	0.0009*	0.0009*	0.0026*	0.0026*	0.0016*

Table 6: Numbers of cases where the results for the dataset *YelpCleanNormal* were worse, better or equivalent to the results for the dataset *YelpClean*.

	C. Reduction	***Weight PoF***	***Filter PoF***
Worse	7 (4%)	8 (4%)	13 (7%)
Better	10 (6%)	8 (4%)	10 (6%)
Equivalent	163 (90%)	164 (91%)	157 (87%)

In some cases the results with the dataset *YelpCleanNormal* were lower, with statistical significance. The cases that did not have statistical difference represent on average 90% of the cases. Thus, text normalization did not result in significant improvements for the recommendation.

In Table 7, we summarize the results by displaying the best configurations for each privileged information. For the dataset *YelpClean*, the algorithm *C. Reduction* presented its best result using the context *Date_1_15_20*. The *Weight PoF* algorithm performed best using the context *OrgTime_05_15_20*. Finally, the algorithm *Filter PoF* presented the best result among all the cases with the context *LocalTime_0_2_7*.

Table 7: The best configurations and values of $MAP@10$ (between parenthesis) for each privileged information, considering the method *TopicAsContext* and the "*Context of Reviews*".

Privileged Information	YelpClean			YelpCleanNormal		
	C. Reduc.	***Weight PoF***	***Filter PoF***	***C. Reduc.***	***Weight PoF***	***Filter PoF***
Date	0_15_20 (0.0029)	0_15_20 (0.0029)	0_10_50 (0.0033)	0_15_20 (0.0070)	0_15_20 (0.0070)	0_15_20 (0.0046)
DateLocal	1_15_20 (0.0086)	1_15_20 (0.0086)	05_15_20 (0.0036)	0_10_50 (0.0038)	0_10_50 (0.0037)	1_50_100 (0.0055)
DateLocal_Org	0_2_7 (0.0044)	05_15_20 (0.0058)	1_15_20 (0.0081)	05_15_20 (0.0047)	05_15_20 (0.0047)	1_2_7 (0.0056)
DateLocalTime	0_15_20 (0.0034)	0_15_20 (0.0034)	05_2_7 (0.0044)	0_2_7 (0.0050)	0_2_7 (0.0051)	0_15_20 (0.0042)
DateOrg	0_15_20 (0.0041)	0_15_20 (0.0041)	05_15_20 (0.0055)	0_50_100 1_2_7 (0.0065)	0_50_100 1_2_7 (0.0066)	1_2_7 (0.0093)
DateOrgTime	1_15_20 (0.0051)	1_15_20 (0.0053)	1_15_20 (0.0078)	1_50_100 (0.0044)	1_50_100 (0.0044)	1_15_20 (0.0045)
DateTime	1_15_20 (0.0061)	1_15_20 (0.0060)	05_50_100 (0.0044)	05_15_20 (0.0070)	05_15_20 (0.0070)	05_15_20 (0.0073)
Local	0_15_20 (0.0033)	0_15_20 05_50_100 (0.0033)	1_10_50 (0.0058)	0_10_50 (0.0032)	0_10_50 (0.0032)	0_2_7 (0.0026)
LocalOrg	1_15_20 (0.0057)	1_15_20 (0.0057)	1_2_7 (0.0038)	0_15_20 (0.0059)	0_15_20 (0.0075)	0_15_20 (0.0061)
LocalOrgTime	0_2_7 (0.0046)	0_15_20 (0.0056)	0_2_7 (0.0048)	05_15_20 (0.0058)	05_15_20 (0.0058)	1_15_20 (0.0037)
LocalTime	0_2_7 (0.0051)	0_2_7 (0.0051)	0_2_7 (0.0096)	0_2_7 (0.0052)	0_2_7 (0.0052)	1_15_20 (0.0053)
Org	05_50_100 (0.0053)	05_50_100 (0.0057)	05_50_100 (0.0059)	05_15_20 (0.0048)	05_15_20 (0.0049)	05_2_7 1_15_20 (0.0055)
OrgTime	05_15_20 (0.0074)	05_15_20 (0.0087)	05_15_20 (0.0080)	1_10_50 (0.0045)	1_15_20 (0.0056)	1_15_20 (0.0060)
Time	0_15_20 (0.0045)	0_15_20 (0.0045)	05_15_20 (0.0026)	0_50_100 (0.0038)	0_50_100 (0.0039)	05_15_20 (0.0029)
DateLocalOrgTime	1_10_50 (0.0051)	1_10_50 (0.0051)	0_2_7 (0.0078)	1_15_20 (0.0072)	1_15_20 (0.0072)	1_15_20 (0.0095)

For the dataset *YelpCleanNormal*, the algorithm *C. Reduction* presented the best result using the context *DateLocalOrgTime_1_50_100*. In contrast, the *Weight PoF* and *Filter PoF* algorithms had their best performances using the contexts *LocalOrg_0_15_20* and *DateLocalOrgTime_1_15_20*, respectively.

Analyzing the characteristics of Table 2, we did not notice a direct relationship between them and the results, that is, the numbers of contexts, transactions or contexts per items apparently did not influence MAP values.

3.5.4 TopicAsContext considering the "Context of Items"

In this case the *YelpCleanNormal* dataset was also not considered, since text normalization did not improve the results considerably. In the same way as in the *EntityAsContext* method, considering contexts by the *TopicAsContext* method as *"Context of Items"* greatly improved the results, especially for the algorithms *C. Reduction* and *Weight PoF*

(Table 8).

In all cases the algorithms *C. Reduction* and *Weight PoF* were better or equivalent to IBCF. Regarding *Filter PoF*, its results were lower than the IBCF results, with statistical significance, in all cases.

Table 8: Comparing the context-aware recommendation algorithms using contexts of the method *TopicAsContext* against the non-contextual baseline IBCF. The values that are statistically different than IBCF (p-value>0.05) are together with a asterisk and the values that are better than IBCF are in boldface (considering the *"Context of Items"*).

Context	**IBCF**	**YelpClean**		
		C. Reduc.	***Weight PoF***	***Filter PoF***
Date_0_2_7	0.0215	**0.0230**	**0.0264***	0.0033*
Date_0_10_50	0.0215	**0.0219**	**0.0228**	0.0051*
Date_0_15_20	0.0215	**0.0272***	**0.0296***	0.0041*
Date_0_50_100	0.0215	**0.0219**	**0.0227**	0.0043*
Date_05_2_7	0.0215	**0.0238***	**0.0272***	0.0025*
Date_05_10_50	0.0215	**0.0218**	**0.0230**	0.0045*
Date_05_15_20	0.0215	**0.0261***	**0.0282***	0.0019*
Date_05_50_100	0.0215	**0.0224**	**0.0230**	0.0042*
Date_1_2_7	0.0215	**0.0221**	**0.0251***	0.0014*
Date_1_10_50	0.0215	**0.0238**	**0.0250***	0.0050*
Date_1_15_20	0.0215	**0.0250***	**0.0270***	0.0067*
Date_1_50_100	0.0215	**0.0233**	**0.0239***	0.0043*
DateLocal_0_2_7	0.0215	**0.0238**	**0.0277***	0.0089*
DateLocal_0_10_50	0.0215	**0.0224**	**0.0235**	0.0043*
DateLocal_0_15_20	0.0215	**0.0253***	**0.0276***	0.0055*
DateLocal_0_50_100	0.0215	**0.0227**	**0.0231**	0.0028*
DateLocal_05_2_7	0.0215	**0.0234**	**0.0269***	0.0044*
DateLocal_05_10_50	0.0215	0.0214	**0.0224**	0.0005*
DateLocal_05_15_20	0.0215	**0.0261***	**0.0279***	0.0094*
DateLocal_05_50_100	0.0215	**0.0223**	**0.0229**	0.0037*
DateLocal_1_2_7	0.0215	**0.0223**	**0.0251***	0.0036*
DateLocal_1_10_50	0.0215	**0.0217**	**0.0225**	0.0083*
DateLocal_1_15_20	0.0215	**0.0251***	**0.0267***	0.0061*
DateLocal_1_50_100	0.0215	**0.0231**	**0.0239**	0.0014*
DateLocalOrg_0_2_7	0.0215	**0.0226**	**0.0264***	0.0049*
DateLocalOrg_0_10_50	0.0215	**0.0228**	**0.0236**	0.0045*
DateLocalOrg_0_15_20	0.0215	**0.0263***	**0.0293***	0.0054*
DateLocalOrg_0_50_100	0.0215	0.0205	0.0212	0.0007*
DateLocalOrg_05_2_7	0.0215	**0.0253***	**0.0289***	0.0106*
DateLocalOrg_05_10_50	0.0215	**0.0234***	**0.0245***	0.0057*
DateLocalOrg_05_15_20	0.0215	**0.0279***	**0.0306***	0.0086*
DateLocalOrg_05_50_100	0.0215	**0.0223**	**0.0229**	0.0037*
DateLocalOrg_1_2_7	0.0215	**0.0245***	**0.0280***	0.0081*
DateLocalOrg_1_10_50	0.0215	**0.0236**	**0.0246***	0.0064*
DateLocalOrg_1_15_20	0.0215	**0.0284***	**0.0310***	0.0087*
DateLocalOrg_1_50_100	0.0215	**0.0218**	**0.0228**	0.0025*
DateLocalTime_0_2_7	0.0215	**0.0233**	**0.0269***	0.0090*
DateLocalTime_0_10_50	0.0215	**0.0230**	**0.0238***	0.0043*
DateLocalTime_0_15_20	0.0215	**0.0267***	**0.0290***	0.0081*

Table 8: (continued)

Context	**IBCF**	**YelpClean**		
		C. Reduc.	***Weight PoF***	***Filter PoF***
DateLocalTime_0_50_100	0.0215	**0.0226**	**0.0233**	0.0037*
DateLocalTime_05_2_7	0.0215	**0.0230**	**0.0263***	0.0085*
DateLocalTime_05_10_50	0.0215	0.0213	**0.0223**	**0.0022***
DateLocalTime_05_15_20	0.0215	**0.0244**	**0.0267***	0.0046*
DateLocalTime_05_50_100	0.0215	0.0161	**0.0230**	0.0052*
DateLocalTime_1_2_7	0.0215	**0.0242**	**0.0273***	0.0073*
DateLocalTime_1_10_50	0.0215	**0.0217**	**0.0225**	0.0057*
DateLocalTime_1_15_20	0.0215	**0.0243**	**0.0259***	0.0098*
DateLocalTime_1_50_100	0.0215	**0.0235**	**0.0243***	0.0062*
DateOrg_0_2_7	0.0215	**0.0230**	**0.0268***	0.0060*
DateOrg_0_10_50	0.0215	**0.0223**	**0.0234**	0.0036*
DateOrg_0_15_20	0.0215	**0.0287***	**0.0310***	0.0096*
DateOrg_0_50_100	0.0215	0.0210	**0.0218**	0.0034*
DateOrg_05_2_7	0.0215	**0.0252***	**0.0298***	0.0077*
DateOrg_05_10_50	0.0215	**0.0222**	**0.0232**	0.0041*
DateOrg_05_15_20	0.0215	**0.0276***	**0.0303***	0.0103*
DateOrg_05_50_100	0.0215	**0.0224**	**0.0231**	0.0009*
DateOrg_1_2_7	0.0215	**0.0253***	**0.0284***	0.0113*
DateOrg_1_10_50	0.0215	**0.0236**	**0.0245***	0.0077*
DateOrg_1_15_20	0.0215	**0.0271***	**0.0297***	0.0089*
DateOrg_1_50_100	0.0215	**0.0240***	**0.0249***	0.0039*
DateOrgTime_0_2_7	0.0215	**0.0244***	**0.0277***	0.0084*
DateOrgTime_0_10_50	0.0215	**0.0235***	**0.0245***	0.0046*
DateOrgTime_0_15_20	0.0215	**0.0271***	**0.0299***	0.0061*
DateOrgTime_0_50_100	0.0215	**0.0225**	**0.0230**	0.0036*
DateOrgTime_05_2_7	0.0215	**0.0239***	**0.0273***	0.0099*
DateOrgTime_05_10_50	0.0215	0.0207	**0.0216**	0.0043
DateOrgTime_05_15_20	0.0215	**0.0235**	**0.0258***	0.0039*
DateOrgTime_05_50_100	0.0215	0.0212	**0.0219**	0.0043*
DateOrgTime_1_2_7	0.0215	**0.0231**	**0.0263***	0.0074*
DateOrgTime_1_10_50	0.0215	**0.0218**	**0.0228**	0.0053*
DateOrgTime_1_15_20	0.0215	**0.0260***	**0.0283***	0.0065*
DateOrgTime_1_50_100	0.0215	0.0211	**0.0221**	0.0009*
DateTime_0_2_7	0.0215	**0.0235***	**0.0267***	0.0091*
DateTime_0_10_50	0.0215	**0.0234***	**0.0240***	0.0015*
DateTime_0_15_20	0.0215	**0.0265***	**0.0287***	0.0082*
DateTime_0_50_100	0.0215	**0.0221**	**0.0228**	0.0040*
DateTime_05_2_7	0.0215	**0.0239**	**0.0273***	0.0029*
DateTime_05_10_50	0.0215	0.0212	**0.0222**	0.0012*
DateTime_05_15_20	0.0215	**0.0248**	**0.0267***	0.0036*
DateTime_05_50_100	0.0215	**0.0226**	**0.0235**	0.0049*
DateTime_1_2_7	0.0215	**0.0233**	**0.0260***	0.0014*
DateTime_1_10_50	0.0215	0.0209	**0.0219**	0.0008*
DateTime_1_15_20	0.0215	**0.0240***	**0.0260***	0.0039*
DateTime_1_50_100	0.0215	**0.0216**	**0.0225**	0.0043*
Local_0_2_7	0.0215	**0.0225**	**0.0260***	0.0092*
Local_0_10_50	0.0215	**0.0216**	**0.0230**	0.0054*
Local_0_15_20	0.0215	**0.0250***	**0.0276***	0.0094*
Local_0_50_100	0.0215	**0.0217**	**0.0224**	0.0032*
Local_05_2_7	0.0215	**0.0231**	**0.0270***	0.0056*

Table 8: (continued)

Context	IBCF	YelpClean		
		C. Reduc.	***Weight PoF***	***Filter PoF***
Local_05_10_50	0.0215	**0.0231**	**0.0242***	0.0044*
Local_05_15_20	0.0215	**0.0241**	**0.0262***	0.0042*
Local_05_50_100	0.0215	**0.0220**	**0.0226**	0.0009*
Local_1_2_7	0.0215	**0.0229**	**0.0256***	0.0077*
Local_1_10_50	0.0215	**0.0244***	**0.0256***	0.0085*
Local_1_15_20	0.0215	**0.0259***	**0.0280***	0.0107*
Local_1_50_100	0.0215	**0.0219**	**0.0227**	0.0052*
LocalOrg_0_2_7	0.0215	**0.0249***	**0.0288***	0.0134*
LocalOrg_0_10_50	0.0215	**0.0236**	**0.0245***	0.0049*
LocalOrg_0_15_20	0.0215	**0.0261***	**0.0289***	0.0078*
LocalOrg_0_50_100	0.0215	**0.0228**	**0.0239**	0.0035*
LocalOrg_05_2_7	0.0215	**0.0235**	**0.0273***	0.0089*
LocalOrg_05_10_50	0.0215	**0.0227**	**0.0237**	0.0034*
LocalOrg_05_15_20	0.0215	**0.0257***	**0.0281***	0.0059*
LocalOrg_05_50_100	0.0215	**0.0226**	**0.0233***	0.0033*
LocalOrg_1_2_7	0.0215	**0.0238**	**0.0270***	0.0084*
LocalOrg_1_10_50	0.0215	**0.0230**	**0.0240**	0.0090
LocalOrg_1_15_20	0.0215	**0.0262***	**0.0285***	0.0109*
LocalOrg_1_50_100	0.0215	**0.0234**	**0.0242**	0.0044*
LocalOrgTime_0_2_7	0.0215	**0.0226**	**0.0262***	0.0089*
LocalOrgTime_0_10_50	0.0215	**0.0222**	**0.0231**	0.0045*
LocalOrgTime_0_15_20	0.0215	**0.0259***	**0.0286***	0.0104*
LocalOrgTime_0_50_100	0.0215	**0.0228**	**0.0234**	0.0032*
LocalOrgTime_05_2_7	0.0215	**0.0219**	**0.0257***	0.0039*
LocalOrgTime_05_10_50	0.0215	0.0215	**0.0223**	0.0043*
LocalOrgTime_05_15_20	0.0215	**0.0246***	**0.0275***	0.0069*
LocalOrgTime_05_50_100	0.0215	0.0214	**0.0221**	0.0011*
LocalOrgTime_1_2_7	0.0215	**0.0237**	**0.0267***	0.0076*
LocalOrgTime_1_10_50	0.0215	**0.0232**	**0.0242***	0.0059*
LocalOrgTime_1_15_20	0.0215	**0.0253***	**0.0276***	0.0070*
LocalOrgTime_1_50_100	0.0215	**0.0234**	**0.0242**	0.0044*
LocalTime_0_2_7	0.0215	**0.0224**	**0.0256***	0.0089*
LocalTime_0_10_50	0.0215	**0.0244***	**0.0255***	0.0042*
LocalTime_0_15_20	0.0215	**0.0253***	**0.0276***	0.0051*
LocalTime_0_50_100	0.0215	**0.0236**	**0.0243**	0.0042*
LocalTime_05_2_7	0.0215	0.0205	**0.0240***	0.0085*
LocalTime_05_10_50	0.0215	**0.0217**	**0.0228**	0.0041*
LocalTime_05_15_20	0.0215	**0.0246**	**0.0275***	0.0087*
LocalTime_05_50_100	0.0215	**0.0217**	**0.0227**	0.0048*
LocalTime_1_2_7	0.0215	**0.0235***	**0.0266***	0.0079*
LocalTime_1_10_50	0.0215	**0.0231***	**0.0237***	0.0077*
LocalTime_1_15_20	0.0215	**0.0239***	**0.0256***	0.0029*
LocalTime_1_50_100	0.0215	**0.0227***	**0.0234***	0.0058*
Org_0_2_7	0.0215	**0.0245***	**0.0276***	0.0112*
Org_0_10_50	0.0215	**0.0217**	**0.0226**	0.0041*
Org_0_15_20	0.0215	**0.0273***	**0.0296***	0.0093*
Org_0_50_100	0.0215	**0.0222**	**0.0231**	0.0030*
Org_05_2_7	0.0215	**0.0233**	**0.0276***	0.0115*
Org_05_10_50	0.0215	**0.0231**	**0.0243**	0.0055*
Org_05_15_20	0.0215	**0.0285***	**0.0321***	0.0079*

Table 8: (continued)

Context	IBCF	YelpClean		
		C. Reduc.	*Weight PoF*	*Filter PoF*
Org_05_50_100	0.0215	**0.0217**	**0.0226**	0.0006*
Org_1_2_7	0.0215	**0.0242***	**0.0285***	0.0121*
Org_1_10_50	0.0215	**0.0222**	**0.0231***	0.0007*
Org_1_15_20	0.0215	**0.0262***	**0.0289***	0.0037*
Org_1_50_100	0.0215	**0.0220**	**0.0228**	0.0011*
OrgTime_0_2_7	0.0215	**0.0247**	**0.0277***	0.0104*
OrgTime_0_10_50	0.0215	**0.0223**	**0.0234**	0.0047*
OrgTime_0_15_20	0.0215	**0.0286***	**0.0314***	0.0037*
OrgTime_0_50_100	0.0215	**0.0230**	**0.0238**	0.0037*
OrgTime_05_2_7	0.0215	**0.0232**	**0.0275***	0.0049*
OrgTime_05_10_50	0.0215	**0.0226**	**0.0237**	0.0028*
OrgTime_05_15_20	0.0215	**0.0277***	**0.0308***	0.0089*
OrgTime_05_50_100	0.0215	**0.0222**	**0.0232**	0.0034*
OrgTime_1_2_7	0.0215	**0.0248***	**0.0282***	0.0109*
OrgTime_1_10_50	0.0215	**0.0239**	**0.0249**	0.0063*
OrgTime_1_15_20	0.0215	**0.0240**	**0.0265***	0.0020*
OrgTime_1_50_100	0.0215	**0.0233**	**0.0241**	0.0011*
Time_0_2_7	0.0215	**0.0230**	**0.0264***	0.0072*
Time_0_10_50	0.0215	**0.0219**	**0.0227**	0.0045*
Time_0_15_20	0.0215	**0.0258***	**0.0284***	0.0029*
Time_0_50_100	0.0215	**0.0217**	**0.0222**	0.0039*
Time_05_2_7	0.0215	**0.0242***	**0.0271***	0.0068*
Time_05_10_50	0.0215	**0.0225**	**0.0234**	0.0042*
Time_05_15_20	0.0215	**0.0253***	**0.0271***	0.0056*
Time_05_50_100	0.0215	**0.0217**	**0.0226**	0.0048*
Time_1_2_7	0.0215	0.0213	**0.0237**	0.0062*
Time_1_10_50	0.0215	0.0211	**0.0221**	0.0055*
Time_1_15_20	0.0215	**0.0239**	**0.0258***	0.0060*
Time_1_50_100	0.0215	**0.0233**	**0.0239***	0.0043*
DateLocalOrgTime_0_2_7	0.0215	**0.0235***	**0.0274***	0.0094*
DateLocalOrgTime_0_10_50	0.0215	**0.0229**	**0.0237**	0.0040*
DateLocalOrgTime_0_15_20	0.0215	**0.0301***	**0.0301***	0.0020*
DateLocalOrgTime_0_50_100	0.0215	0.0213	**0.0218**	0.0033*
DateLocalOrgTime_05_2_7	0.0215	**0.0223**	**0.0232**	0.0070*
DateLocalOrgTime_05_10_50	0.0215	**0.0237**	**0.0267***	0.0087*
DateLocalOrgTime_05_15_20	0.0215	**0.0224**	**0.0233**	0.0049*
DateLocalOrgTime_05_50_100	0.0215	**0.0254***	**0.0278***	0.0054*
DateLocalOrgTime_1_2_7	0.0215	**0.0241***	**0.0266***	0.0077*
DateLocalOrgTime_1_10_50	0.0215	**0.0225**	**0.0235**	0.0028*
DateLocalOrgTime_1_15_20	0.0215	**0.0271***	**0.0296***	0.0039*
DateLocalOrgTime_1_50_100	0.0215	**0.0223**	**0.0232**	0.0070*

In Table 9, we present the best configurations for each privileged information. For the algorithm *C. Reduction*, the best result was obtained using the context *DateLocalOrgTime_0_15_20*. For the *Weight PoF* algorithm, the context *Org_05_15_20* was the one that generated the best result. For the algorithm *Filter PoF*, the best result was obtained using the context *LocalOrg_0_2_7*.

Table 9: The best configurations and values of $MAP@10$ (between parenthesis) for each privileged information, considering the method *TopicAsContext* and the *"Context of Items"*.

Privileged Information	**YelpClean**		
	C. Reduc.	***Weight PoF***	***Filter PoF***
Date	0_15_20 (0.0272)	0_15_20 (0.0296)	1_15_20 (0.0067)
DateLocal	05_15_20 (0.0261)	05_15_20 (0.0279)	05_15_20 (0.0094)
DateLocalOrg	1_15_20 (0.0284)	1_15_20 (0.0310)	05_2_7 (0.0106)
DateLocalTime	0_15_20 (0.0267)	0_15_20 (0.0290)	1_15_20 (0.0098)
DateOrg	0_15_20 (0.0287)	0_15_20 (0.0310)	1_2_7 (0.0113)
DateOrgTime	0_15_20 (0.0271)	0_15_20 (0.0299)	05_2_7 (0.0099)
DateTime	0_15_20 (0.0265)	0_15_20 (0.0287)	0_2_7 (0.0091)
Local	1_15_20 (0.0259)	1_15_20 (0.0280)	1_15_20 (0.0107)
LocalOrg	1_15_20 (0.0262)	0_15_20 (0.0289)	0_2_7 (0.0134)
LocalOrgTime	0_2_7 (0.0259)	0_2_7 (0.0286)	0_2_7 (0.0104)
LocalTime	0_15_20 (0.0253)	0_15_20 (0.0276)	0_2_7 (0.0089)
Org	05_15_20 (0.0285)	05_15_20 (0.0321)	1_2_7 (0.0121)
OrgTime	0_15_20 (0.0286)	0_15_20 (0.0314)	1_2_7 (0.0109)
Time	0_15_20 (0.0258)	0_15_20 (0.0284)	0_2_7 (0.0072)
DateLocalOrgTime	0_15_20 (0.0301)	0_15_20 (0.0301)	0_2_7 (0.0094)

In Figure 6, some of the best results of the method *TopicAsContext* are presented in three graphs. The first graph shows the results for the dataset *YelpClean* using the *"Context of Reviews"*. The second graph shows the results for the dataset *YelpCleanNormal* using the *"Context of Reviews"*. And finally, in the third we look at the results for the dataset *YelpClean* using the *"Context of Items"*. Analyzing the graphs, we note that: the results for the dataset *YelpClean* were higher than the results of the dataset *YelpCleanNormal* and the results using the *"Context of Items"* were higher than the results when the *"Context of Reviews"* was used.

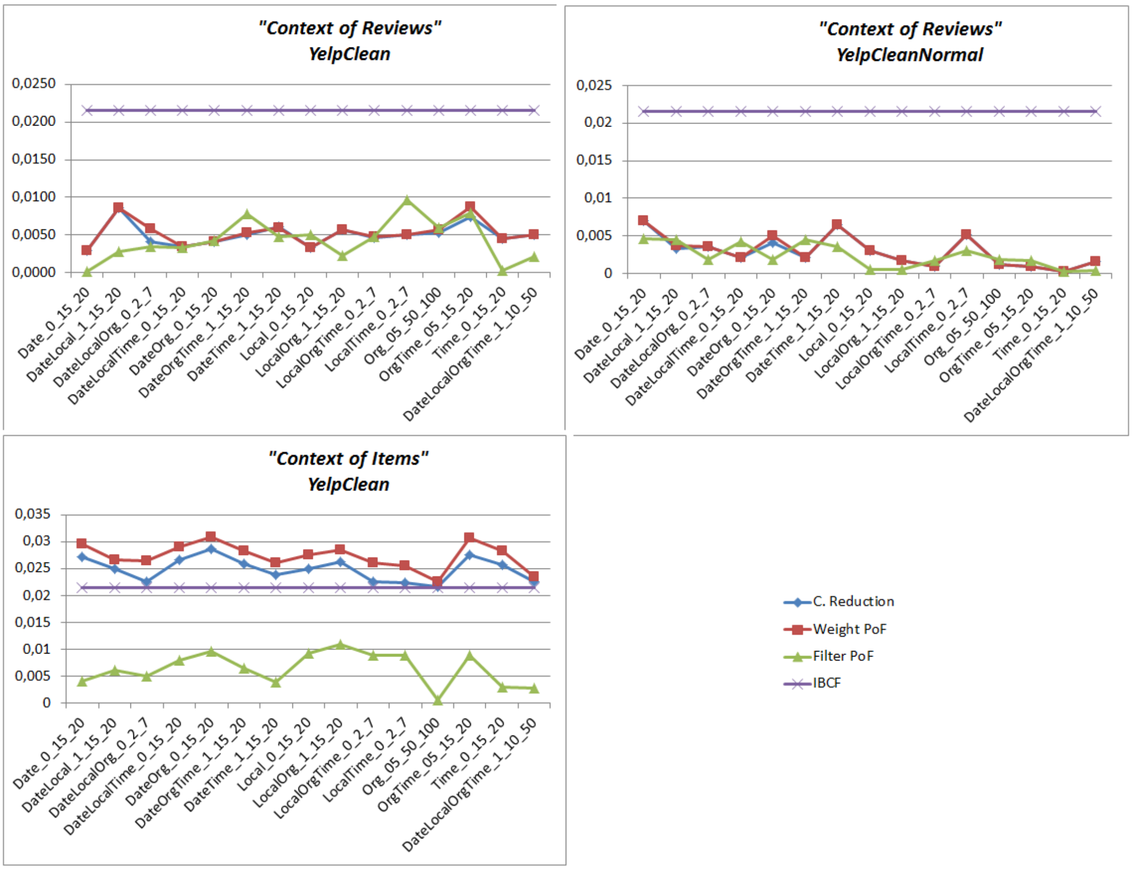

Figure 6: Graphics with some of the best results of the method *TopicAsContext*.

4. Final Remarks

Recommender systems are systems that recommend items that may be interesting to users. Traditional systems consider only item and user information to generate recommendations. However, the use of additional information, such as contextual information, may result in more accurate recommendations. In this sense there is a class of recommender systems, called context-aware recommender systems, that generates recommendations using information from users and items, as well as contextual information.

Work in the area of context-aware recommender systems has shown that the use of the contextual information improves the accuracy of the recommendation. However, there is a difficulty in extracting such information. There is a lack of automatic and effective extraction methods. In addition, it is necessary to define the best source for extracting relevant contextual information.

With the advent of Web 2.0, users have generated a lot of their own content through reviews, posts on social networks, etc. Such content is rich in information about the user context and opinion. From the textual content of reviews, we can extract a lot of information that can be useful in the recommendation process.

Some works have already been developed with the intention of proposing methods for the contextual information extraction. Two of these methods were proposed and applied in the Web page domain. One of them, the *EntityAsContext* method, proposed by Domingues et al. (2014), consists of extracting named entities from the textual content of web pages and using them as contextual information in recommender systems. The second method, *TopicAsContext*, proposed by Sundermann et al. (2016), consists of building topic hierarchies of web pages, extracting and using the topics as contextual information.

The objective of this work was to apply the two methods previously mentioned in the domain of reviews and to evaluate the recommendations generated in this scenario, building, in this way, baselines for future works. For this purpose, the Yelp dataset made available for the ACM RecSysChallenge 2013 was used. The review texts were extracted, passed through a cleaning process and also normalized, generating the *YelpClean* and *YelpCleanNormal* datasets. The evaluation consisted of comparing the values of the MAP metric, obtained by four context-aware recommender systems, against the MAP values obtained by the IBCF method, that does not use context. The constextual information that was fed into the context-aware recommender systems was extracted by the "*EntityAsContext*" and "*TopicAsContext*" methods.

Results were presented and discussed, taking into account the two methods (*EntityAs-*

Context" and "*TopicAsContext*"), the datasets and the two ways of considering contextual information. The text normalization did not improve the quality of the contexts extracted, as there was no statistically significant improvement in the performance of the recommendation. In addition, the *"Context of Items"* generated more precise recommendations than the *"Context of Reviews"*. Finally, the "*EntityAsContext*" method outperformed the "*TopicAsContext*" method'.

Concluding, in this work we analyzed the performance of methods already proposed in the literature being applied in the domain of reviews. The results of the evaluation can be used in other works as baselines for new methods. In addition, this work, as well as the discussion of the results, can inspire and assist in other work in development.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq) - grant #403648/2016-5, and Fundação Araucária de Apoio ao Desenvolvimento Científico e Tecnológico do Estado do Paraná - Brasil (FAPPR).

References

Adomavicius, G., R. Sankaranarayanan, S. Sen, & A. Tuzhilin (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems 23*(1), 103–145.

Adomavicius, G. & A. Tuzhilin (2005). 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.

Adomavicius, G. & A. Tuzhilin (2011). Context-aware recommender systems. In *Recommender Systems Handbook*, pp. 217–253. Springer.

Bauman, K. & A. Tuzhilin (2014). Discovering contextual information from user reviews for recommendation purposes. In *CBRecSys'14: Proceedings of Workshop on New Trends in Content-based Recommender Systems*, pp. 2–9.

Biancalana, C., F. Gasparetti, A. Micarelli, & G. Sansonetti (2013, February). An approach to social recommendation for context-aware mobile services. *ACM Transactions on Intelligent Systems and Technology 4*(1), 10:1–10:31.

Breese, J. S., D. Heckerman, & C. Kadie (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *UAI'98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 43–52.

Cardoso, N. (2008). Rembrandt - reconhecimento de entidades mencionadas baseado em relaç oes e análise detalhada do texto. In *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, pp. 195–211. Linguateca.

Chen, G. & L. Chen (2014). Recommendation based on contextual opinions. In *User Modeling, Adaptation, and Personalization*, Volume 8538 of *Lecture Notes in Computer Science*, pp. 61–73. Springer International Publishing.

Chen, G. & L. Chen (2015). Augmenting service recommender systems by incorporating contextual opinions from user reviews. *User Modeling and User-Adapted Interaction 25*(3), 295–329.

Conrado, M. S. (2014). *Extração automática de termos simples baseada em aprendizado de máquina.* Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo - USP.

Conrado, M. S., T. A. S. Pardo, & S. O. Rezende (2013). A machine learning approach to automatic term extraction using a rich feature set. In *NAACL-HLT-SRW'13: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Student Research Workshop*, Atlanta, Georgia, pp. 16–23. Association for Computational Linguistics.

Cunningham, H., D. Maynard, K. Bontcheva, & V. Tablan (2002). Gate: an architecture for development of robust hlt applications. In *ACL'02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, pp. 168–175. Association for Computational Linguistics.

Deshpande, M. & G. Karypis (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems 22*(1), 143–177.

Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing 5*(1), 4–7.

Domingues, M. A., A. M. Jorge, & C. Soares (2013). Dimensions as virtual items: Improving the predictive ability of top-n recommender systems. *Information Processing and Management: an International Journal 49*(3), 698–720.

Domingues, M. A., M. G. Manzato, R. M. Marcacini, C. V. Sundermann, & S. O. Rezende (2014). Using contextual information from topic hierarchies to improve context-aware recommender systems. In *ICPR'14: Proceedings of the 22nd International Conference on Pattern Recognition*, pp. 3606–3611.

Domingues, M. A., C. V. Sundermann, M. G. Manzato, R. M. Marcacini, & S. O. Rezende (2014). Exploiting text mining techniques for contextual recommendations. In *WI'14: IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, Volume 2, pp. 210–217.

Everitt, B. S., S. Landau, M. Leese, & D. Stahl (2011). *Cluster Analysis* (5th ed.). London, UK: Wiley Publishing.

Finkel, J. R., T. Grenager, & C. Manning (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL'05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, pp. 363–370. Association for Computational Linguistics.

Hariri, N., B. Mobasher, R. Burke, & Y. Zheng (2011). Context-aware recommendation based on review mining. In *ITWP'11: Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, pp. 30–36.

Jones, S. & G. W. Paynter (2002). Automatic extraction of document keyphrases for use in digital libraries: Evaluation and applications. *Journal of the American Society for Information Science and Technology 53*(8), 653–677.

Lafferty, J. D., A. McCallum, & F. C. N. Pereira (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'01: Proceedings of the Eighteenth International Conference on Machine Learning*, San Francisco, CA, USA, pp. 282–289. Morgan Kaufmann Publishers Inc.

Lee, D., S. E. Park, M. Kahng, S. Lee, & S. G. Lee (2010). Exploiting contextual information from event logs for personalized recommendation. In *Computer and Information Science 2010*, Volume 317 of *Studies in Computational Intelligence*, pp. 121–139. Yamagata, Japan: Springer Berlin Heidelberg.

Levi, A., O. Mokryn, C. Diot, & N. Taft (2012). Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *RecSys'12: Proceedings of the Sixth ACM Conference on Recommender Systems*, New York, NY, USA, pp. 115–122. ACM.

Li, Y., J. Nie, & Y. Zhang (2010). Contextual recommendation based on text mining. In *COLING'10: Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 692–700.

Linden, G., B. Smith, & J. York (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing 7*(1), 76–80.

Marcacini, R. & S. O. Rezende (2013). Incremental hierarchical text clustering with privileged information. In *DocEng'13: Proceedings of the 2013 ACM Symposium on Document Engineering*, pp. 231–232.

Marcacini, R. M. & S. O. Rezende (2010). Torch: a tool for building topic hierarchies from growing text collections. In *WebMedia'10: 8th Brazilian Symposium on Multimedia and the Web - IX Workshop on Tools and Applications.*

Panniello, U. & M. Gorgoglione (2012). Incorporating context into recommender systems: An empirical comparison of context-based approaches. *Eletronic Commerce Research 12*(1), 1–30.

Ricci, F., L. Rokach, B. Shapira, & P. B. Kantor (Eds.) (2011). *Recommender System Handbook*. Springer.

Rossi, R. G. (2011). Representação de coleções de documentos textuais por meio de regras de associaccão. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo - USP.

Sekine, S. (2004). Named Entity: History and Future.

Sundermann, C., M. Domingues, R. Marcacini, & S. Rezende (2014). Using topic hierarchies with privileged information to improve context-aware recommender systems. In *BRACIS'14: Proceedings of Brazilian Conference on Intelligent Systems*, pp. 61–66.

Sundermann, C. V., J. Antunes, M. A. Domingues, & S. O. Rezende (2018). Exploration of word embedding model to improve context-aware recommender systems. In *WI'18: IEEE/WIC/ACM International Conference on Web Intelligence*.

Sundermann, C. V., M. A. Domingues, M. da Silva Conrado, & S. O. Rezende (2016). Privileged contextual information for context-aware recommender systems. *Expert Systems with Applications 57*, 139 – 158.

Sundermann, C. V., M. A. Domingues, R. M. Marcacini, & S. O. Rezende (2015). Combining privileged information to improve context-aware recommender systems. In *ENIAC'15: Encontro Nacional de Inteligência Artificial e Computacional*. [Online]. Available: https://arxiv.org/abs/1511.02290.

Takehara, T., S. Miki, N. Nitta, & N. Babaguchi (2012). Extracting context information from microblog based on analysis of online reviews. In *ICME'12: IEEE International Conference on Multimedia and Expo Workshops*, pp. 248–253.