Combining Privileged Information to Improve Context-Aware Recommender Systems

Camila V. Sundermann*, Marcos A. Domingues*, Ricardo M. Marcacini† and Solange O. Rezende*

*Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos, SP, Brazil
{camilavs, mad, solange}@icmc.usp.br

†Universidade Federal do Mato Grosso do Sul
Três Lagoas, MS, Brazil
ricardo.marcacini@ufms.br

***Abstract*—A recommender system is an information filtering technology which can be used to predict preference ratings of items (products, services, movies, etc) and/or to output a ranking of items that are likely to be of interest to the user. Context-aware recommender systems (CARS) learn and predict the tastes and preferences of users by incorporating available contextual information in the recommendation process. One of the major challenges in context-aware recommender systems research is the lack of automatic methods to obtain contextual information for these systems. Considering this scenario, in this paper, we propose to use contextual information from topic hierarchies of the items (web pages) to improve the performance of context-aware recommender systems. The topic hierarchies are constructed by an extension of the LUPI-based Incremental Hierarchical Clustering method that considers three types of information: traditional bag-of-words (technical information), and the combination of named entities (privileged information I) with domain terms (privileged information II). We evaluated the contextual information in four context-aware recommender systems. Different weights were assigned to each type of information. The empirical results demonstrated that topic hierarchies with the combination of the two kinds of privileged information can provide better recommendations.**

Keywords*—Contextual Information; Context-Aware Recommender Systems; Text Mining; Topic Hierarchy; Named Entities; Domain Terms

I. Introduction

A recommender system is an information filtering technology which can be used to predict preference ratings of items (products, services, movies, etc) and/or to output a ranking of items that are likely to be of interest to the user [21]. This kind of system has emerged in order to reduce the difficulty of users to choose the product or service that most meets their needs. Many areas have been using recommender systems, mainly some web sites, like Amazon[1], Netflix[2] and Last.fm[3].

Recommender systems usually use web access logs which represent the interaction activity between users and items. Traditional recommender systems consider only the two entities, items and users, to build the recommendation model. However, the use of contextual information can improve the recommendation process in some cases [2], [12]. The researchers that already investigated the use of context discovered that the quality of recommendations increases when additional information, like time, place, and so on, is used.

The concept *context* can assume different definitions. In this paper we consider that context is any information that can be used to characterize the situation of an entity [11]. An example of application in which to consider contextual information can be important is movie recommendation. An user can prefer watch a love story with his girlfriend on Saturday night and a comedy with his friends during the week. So an online video store can recommend the movie that more corresponds to the users context.

Although the proven importance of the use of contextual information in the recommendation process, there is still a lack of automatic methods to obtain such information. In a context-aware recommender system is possible to consider the context of the user or the context of the item. In this work we focus on the context extracted for the items (in our case, web pages).

The contextual information can be represented and structured in various ways. A form of organizing this information is using hierarchical structures. In [2] and [20], the researchers represented the context as trees. Given this possibility of hierarchical organization of context, we have been using topic hierarchies as a way to organize and extract the context of the textual content of web pages [13], [24].

Most of the methods in the literature to build topic hierarchies represent the texts as a traditional bag-of-words, i.e., these methods consider the terms of texts as a disordered set of words. In [13], we constructed topic hierarchies of web pages by using traditional bag-of-words, and the extracted topics were used as context of these pages in context-aware recommender systems. However, Marcacini and Rezende [17] proposed a method, called LUPI-based Incremental Hierarchical Clustering (LIHC) to construct topic hierarchies that uses besides the bag-of-words (technical information), also the privileged information, which is a more valuable kind of information extracted from texts. In [24], we constructed topic

[1] http://www.amazon.com

[2] http://www.netflix.com

[3] http://www.last.fm

hierarchies of the web pages using the method LIHC. We considered the bag-of-words and the named entities, extracted from the web pages, as privileged information, and we used the topics as the contextual information of the web pages in context-aware recommender systems. However, named entities are one of the various types of information that can be considered as privileged information.

The original LIHC method used only one type of privileged information to construct the topic hierarchies, in other words, it did not work with more than a kind of privileged information at the same time. In this paper we extend the method LIHC to be able to work with two kinds of privileged information, i.e., to construct topic hierarchies using besides the technical information, also other two kinds of information (privileged information). So, we propose to use topic hierarchies constructed by using three kinds of information: bag-of-words (technical information), named entities (privileged information I) and domain terms (privileged information II). The aim of this work is to combine this information and evaluate the impact of the use of the topics extracted from this combination as contextual information in context-aware recommender systems.

This paper is structured as follows: in Section II, we report the related work. In Section III, we present our proposal. We evaluate our proposal in Section IV. And, finally, in Section V, we present conclusion and future work.

II. Related Work

There are three different ways to acquire contextual information: *explicitly*, *implicitly* and *inferred* [3]. The explicit acquisition methods collect the contextual information through direct questions to the users. The implicit acquisition methods get contextual information directly from Web data or environment. The inference methods obtain contextual information using data an text mining techniques. In this paper, we infer context from web pages using text mining techniques. Following, some related works are presented.

In [16], Li et al. proposed methods to extract contextual information from online reviews. They investigated available restaurant review data and four types of contextual information for a meal: the company (if the meal involved multiple people), occasion (for which occasions is the event), time (what time of the day) and location (in which city the event took place). They developed their algorithms by using existing natural language processing tools such as GATE tool[4]. Hariri et al. [14] introduced a context-aware recommendation system that obtains contextual information by mining hotel reviews made by users, and combine them with user's rating historic to calculate a utility function over a set of items. They used a hotel review dataset from "*Trip Advisor website*"[5].

The methods proposed by Li et al. [16] and Hariri et al. [14] assume there are explicit contextual information in reviews, and such information is obtained for each review by mapping it to the labels. Therefore, they use supervised methods to learn the labels. The advantage of our proposal is that it exploits unsupervised methods to learn topic hierarchies. Therefore, it does not need a mapping between reviews and labels.

Aciar [1] proposed a technique to detect sentences of reviews with contextual information. She applied text mining tools to define sets of rules for identifying such sentences with context. In her work the phrases are classified into two categories: "Contextual" and "Preferences". The category "Contextual" groups phrases that present information on the context in which the review was written. The category "Preferences" groups phrases that present information about the features that consumers evaluated.

Our work differs from the Aciar's method since it is capable of using more text mining techniques, and these techniques are unsupervised, to extract contextual information. Aciar uses supervised techniques and conduct the evaluation of her method by using a case study, i.e., she does not compare her method results against other methods in the literature. Besides, she does not discuss the use of the extracted information in the recommendation process.

Ho et al. [15] proposed an approach to mine future spatiotemporal events from news articles, and thus provide information for location-aware recommendation systems. A future event consists of its geographic location, temporal pattern, sentiment variable, news title, key phrase, and news article URL. Besides that, their method is unsupervised and also extracts topics.

In [15], the contextual information that Ho et al. extracted are related to time and local. The information of time is extracted from the timestamp of the article publication. To extract information of local, they also used named entity recognition. However, they did not evaluate the impact of the contextual information that they extracted in the recommender systems. The authors only presented some results about the evaluation of the context extraction process.

Bauman and Tuzhilin [4] presented a method to find relevant contextual information from reviews of users. In this method, the reviews are classified as "specific" and "generic". They found that contextual information is contained mainly in the specific reviews, which are those that describe specific visits of a user to an establishment. Therefore, the context is extracted from the "specific" reviews by means of two methods: "word-based" and "LDA-based".

In [4], Bauman and Tuzhilin consider that the contextual information is not known a priori. Besides that, their method is unsupervised and also extracts topics. Our method differs from theirs since it extracts topics using also privileged information, which enrich the contextual information.

Our method has many advantages over the other ones proposed in the literature. In general, it does not need previous information (for example, labels). It uses unsupervised methods and combines technical information with privileged information, which enriches the contextual information. Additionally, the context extracted is about the item (web pages) and not the user. Finally, our results, presented in Section IV, demonstrate that our contextual information is able to improve the quality of recommendations.

III. Our Proposal

As already stated, the term context can assume many definitions depending on what area it is being treated in. We consider

[4] http://gate.ac.uk

[5] http://www.tripadvisor.com

the definition given by Dey [11] that says: “Context is any definition that can be used to characterize the situation of an entity”. In our work the entities are web pages (items). Besides the definition, the contextual information can be represented using many structures. Some researchers treat the context as a hierarchical structure and represent it using trees. For example, Panniello and Gorgoglione [20] represent the attribute “period of year” as a tree like illustrated in Figure 1.

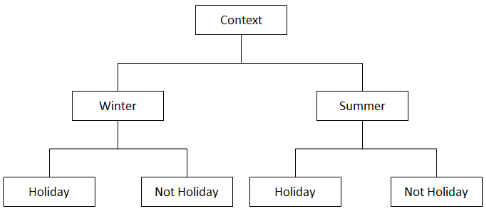

Fig. 1: Hierarchical structure of the contextual attribute “period of the year” [20].

The idea of this research is representing the contextual information using a hierarchical structure called topic hierarchy. Topic hierarchies organize texts into groups and subgroups, and for each group, topics are extracted to represent the main issue of the group. Constructing a topic hierarchy of the items in a recommender system means grouping them by context, i.e., the topics extracted for each group represent the context of the group. Items of a same group are in the same context. We construct topic hierarchies by using the textual content of the web pages, and use the topics as contextual information in context-aware recommender systems.

Topics hierarchies can be constructed using hierarchical clustering. Traditional methods represent the textual collection as a bag-of-words [22], also known as technical information [25]. However, we can extract concepts from the texts that are not represented in a simple bag-of-words. Named entities and domain terms are good examples of concepts that may be formed by a word or by more than a word and that are identified and extracted by using more advanced text preprocessing techniques. Thus, these two kinds of information, named entity and domain term, are considered in this paper as privileged information.

The term Named Entity was born in the Message Understanding Conferences (MUC) and includes names of people, organizations and locations, besides numeric expressions like time, date, money and percent expressions [23]. The named entity recognition is a task that involves identifying words or expressions that belong to categories of named entities [19]. For example, in the sentence: “Ana Maria works at Petrobras, in Brazil, since 1989”. “Ana Maria” is recognized as a person, “Petrobras” as an organization, “Brazil” as a location and “1989” as a date.

Despite the importance of term extraction task, there is still no consensus on the formal definition of what the “term” is. A definition widely accepted of term is given by Cabré and Vivaldi [6], which is: “terminological unit obtained from specialized domain”. In most researches found in literature, the authors state that terms are generally nominal units, since they describe concepts. For example, in the Ecologic domain, the terms “climate”, “plant”, “Atlantic forest” and “soil moisture” are examples of domain terms [8]. The terms are used in applications such as information retrieval, information extraction and summarization.

In our proposal, we instantiate the LUPI-based Incremental Hierarchical Clustering (LIHC) method [17] to construct topic hierarchies using one type of privileged information and technical information. Let $D^{pri} = \{d_1^p, \ldots, d_m^p\}$ and $D^{tec} = \{d_1^t, \ldots, d_m^t, d_{m+1}^t, \ldots, d_n\}$ the sets of documents represented by the privileged information (totaling m documents) and with technical information (totaling n documents), respectively, where $d^p \in D^p$ and $d^t \in D^t$. Note that the number of documents represented by the privileged information, in general, is smaller than the number of documents represented by technical information, i.e, $m \leq n$. This is due to the fact that a significant number of documents do not contain features extracted from privileged information (*e.g.*, named entities and domain terms).

The subset of documents that contain the privileged information and technical information, $Y = \{(d_1^t, d_1^p), \ldots, (d_m^t, d_m^p)\}$, is used for learning the initial clustering model. In this case, various clustering algorithms are run (or repeated runs of the same algorithm with different parameter values) to obtain several clusters from the subset Y. To aggregate the generated clusters, the LIHC method obtains two co-association matrices $M^t(i,j)$ and $M^p(i,j)$ which represent, respectively, the technical information (bag-of-words) clustering model and privileged information clustering model. The combination of these two clustering models is performed by using a consensual co-association matrix:

$$M^F(i,j) = (1-\alpha)M^t(i,j) + \alpha M^p(i,j), \qquad (1)$$

for all items i and j. In this case, the parameter α is a combination factor $(0 \leq \alpha \leq 1)$ that indicates the importance of the privileged information space in the final co-association matrix. The initial model of the LIHC method is obtained by applying any hierarchical clustering algorithm from the matrix M^F. The remaining text documents, i.e., the documents without privileged information, are inserted incrementally into hierarchical clustering by using the nearest neighbor technique. For the construction of topic hierarchies, the topic extraction is based on selection of the most frequent terms of each cluster.

In [24], we constructed topic hierarchies of the web pages by using the method LIHC and considering as privileged information only named entities. In this paper, we construct topic hierarchies by combining named entities and domain terms as privileged information, varying the weight of each type of information. To incorporate the two types of privileged information, the method LIHC was extended as follows.

First, D^{pri} is divided into two sets D^{ne} (the set of the privileged information I named entities) and D^{dt} (the set of the privileged information II domain terms). Let $D^{ne} = \{d_1^{ne}, \ldots, d_r^{ne}\}$ be the set of documents with named entities (totaling r documents) and $D^{dt} = \{d_1^{dt}, \ldots, d_s^{dt}\}$ be the set of documents with domain terms (totaling s documents). Similarly, the matrix $M^p(i,j)$ is divided into two matrices M^{ne} (the named entities clustering model) and M^{dt} (the domain terms clustering model). The combination of the three clustering models (M^t, M^{ne} and M^{dt}) is performed by using the follow consensual co-association matrix:

$$M^{nf}(i,j) = (1-\alpha)M^t(i,j) + \beta M^{ne}(i,j) + \theta M^{dt}(i,j), \quad (2)$$

where, β and θ indicate the importance of the named entities and domain terms, respectively, in the final co-association matrix, and $\beta + \theta = \alpha$. In the next section, we empirically evaluate our proposal by using different values of α, β and θ to construct the topic hierarchies.

IV. Empirical Evaluation

The aim of our work is to study the impact of the context, extracted by our method, in context-aware recommender systems. So, the empirical evaluation consists of comparing the results of the algorithms *C. Reduction* [2], *DaVI-BEST* [12], *Weight PoF* and *Filter PoF* [20], all them using our contextual information, against the uncontextual algorithm *Item-Based Collaborative Filtering* (IBCF) [10]. In this way, we compared the quality of the recommendations generated by using our context against the quality of the recommendations generated without using contextual information. In this section, we present the necessary details to understand our experiments: data set, baseline, context-aware recommender algorithms, experimental setup, evaluation measure and the results.

A. Data Set

In the experiments we used a data set from a Portuguese website about agribusiness that consists of 4,659 users, 15,037 accesses and 1,543 web pages written in Portuguese language. To construct the topic hierarchies for these web pages, we used the textual content of the pages, eliminating header, footer and everything that do not pertaining to the main textual content.

We preprocessed the texts executing traditional text preprocessing tasks: stopword removal and stemming. The representations or "term value matrix" were constructed using the term weighting measure TF-IDF (term frequency-inverse document frequency). Three representations were constructed: the traditional bag-of-words representation (technical information), the named entities representation (privileged information I) and the domain terms representation (privileged information II). We defined the weights testing different combinations of the two kinds of privileged information. The weights are shown in Table I.

We extracted the topics from the topic hierarchies considering three configurations $\{50, 100\}$, $\{15, 20\}$ and $\{2, 7\}$. In this configuration $\{x, y\}$, that represents the granularity level, the parameter x identifies the minimum number of items allowed in the topic, while the parameter y identifies the maximum number of items per topic. Topics with more items associated to them mean topics more generic, while topics with fewer items associated to them mean topics more specific. So, the topics extracted by the configuration $\{50, 100\}$ are more generic and the topics extracted by the configuration $\{2, 7\}$ are more specific. The configuration $\{15, 20\}$ are between the two others, and it was chosen because, in previous experiments, we obtained good results using this granularity level. Besides that, the more generic configuration extracts a lower number of topics, while the more specific configuration extracts a higher number of topics. Therefore, using this configuration we can analyses if the number of topics extracted or the granularity level of this topics influences the quality of the recommendations. In Table II, we can see the number of topics extracted using each configuration.

TABLE I: Values of combination factor α used in this paper and the weights of each information.

Combination Factor	Technical Information	Privileged Information	
		Named Entities	Domain Terms
$\alpha = 0.3$	70%	10%	20%
		20%	10%
		15%	15%
$\alpha = 0.5$	50%	20%	30%
		30%	20%
		25%	25%
$\alpha = 0.7$	30%	20%	50%
		50%	20%
		35%	35%
$\alpha = 1$	0%	30%	70%
		70%	30%
		50%	50%

TABLE II: Extracted Topics

Weights		Granularities		
Named Entities	Domain Terms	$\{50, 100\}$	$\{15, 20\}$	$\{2, 7\}$
10%	20%	46	66	897
20%	10%	42	61	902
15%	15%	44	61	890
20%	30%	46	66	869
30%	20%	49	73	872
25%	25%	51	72	868
20%	50%	45	61	892
50%	20%	46	70	901
35%	35%	42	63	879
30%	70%	39	60	915
70%	30%	35	69	870
50%	50%	46	71	921

B. Supporting Tools and Methods

In the experiments we used JPretext[6] and LIHC[7] for the pre-processing and the hierarchical clustering of the items. These two tools are part of Torch [18], that is a set of tools developed to support text clustering and construction of topic hierarchies. JPretext transforms the collection of texts in a "term value matrix" and LIHC tool implements the LUPI-based Incremental Hierarchical Clustering method.

The named entity recognition was performed by using REMBRANDT [7], a system that recognizes classes of named entities, like things, location, organization, people and others, in texts written in Portuguese. REMBRANDT uses Wikipedia[8] as knowledge base for the classification of the entities.

[6] http://sites.labic.icmc.usp.br/torch/msd2011/jpretext

[7] http://sites.labic.icmc.usp.br/torch/doceng2013

[8] https://www.wikipedia.org

Lastly, for the domain term extraction we used the method MATE-ML (Automatic Term Extraction based on Machine Learning) [9], [8]. This method uses machine learning incorporating rich features of candidate terms. The steps of MATE-ML are: 1) Text Pre-Processing; 2) Extraction of linguistic, statistic and hybrid features; 3) Application of filters; and 4) Generation of inductive models based on machine learning.

C. Baseline

In this paper we considered the un-contextual algorithm *Item-Based Collaborative Filtering* (IBCF) [10] as baseline. Let m be the number of users $U = \{u_1, u_2, ..., u_m\}$ and n the number of items that can be recommended $I = \{i_1, i_2, ..., i_n\}$. An item-based collaborative filtering model M is a matrix representing the similarities among all pairs of items, according to a similarity measure. We used the cosine angle similarity measure, defined as:

$$sim(i_1, i_2) = cos(\overrightarrow{i_1}, \overrightarrow{i_2}) = \frac{\overrightarrow{i_1} \cdot \overrightarrow{i_2}}{\|\overrightarrow{i_1}\| * \|\overrightarrow{i_2}\|}, \quad (3)$$

where $\overrightarrow{i_1}$ and $\overrightarrow{i_2}$ are rating vectors and the operator "$\cdot$" denotes the dot-product of the two vectors. In our case, as we are dealing only with implicit feedback, the rating vectors are binary. The value 1 means that the user accessed the respective item, whereas the value 0 is the opposite.

Given an active user u_a and his set of observable items $O \subseteq I$, the N recommendations are generated as follows. First, we identify the set of candidate items for recommendation R by selecting from the model all items $i \notin O$. Then, for each candidate item $r \in R$, we calculate its recommendation score as:

$$score(u_a, O, r) = \frac{\sum_{i \in K_r \cap O} sim(r, i)}{\sum_{i \in K_r} sim(r, i)}, \quad (4)$$

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 recommendation algorithms used in this work are based on the Item-Based Collaborative Filtering. They are presented in the next section.

D. Context-Aware Recommender Systems

Context-aware recommender systems (CARS) learn and predict the tastes and preferences of users by incorporating available contextual information in the recommendation process. According to Adomavicius and Tuzhilin [3], contextual information can be applied at various stages of the recommendation process. Following this criterion, these systems can be divided into three categories: contextual pre-filtering, contextual modeling and contextual post-filtering.

In this work, we evaluate the effects of using the contextual information, obtained from topic hierarchies, in four different context-aware recommender systems:

- ***C. Reduction*** [2] (Pre-filtering approach): in pre-filtering approaches the contextual information is used as a label for filtering out those data that do not correspond to the specified contextual information. The remaining data that passed the filter (contextualized data) is used to generate the model. *C. Reduction* uses the contextual information as label to segment the data. A recommendation method is run for each contextual segment to determine which segment outperforms the traditional un-contextual recommendation model. The best contextual model is chosen to make the recommendation. Here the best model is the one that has the highest F1 measure.
- ***DaVI-BEST*** [12] (Contextual modeling approach): in this approach the context is used in the recommendation model, i.e., the contextual information is part of the model together with user and item data. *DaVI-BEST* considers the contextual information as virtual items, using them along with the actual items in the recommendation model. All contextual information are evaluated and it is selected the dimension which better outperforms the traditional un-contextual recommendation model to make contextual recommendations.
- ***Weight PoF*** **and** ***Filter PoF*** [20] (Contextual post-filtering approaches): these approaches use the contextual information to reorder and filter out the recommendation, respectively. Firstly, they apply the traditional algorithm to build the un-contextual recommendation model, ignoring the contextual information. Then, the probability of users to access the items given the right context is calculated. This probability is multiplied by scores of items to reorder the recommendations (*Weight PoF*) or is used as a threshold to filter them (*Filter PoF*).

E. Experimental Setup and Evaluation Measures

The protocol considered in this paper to measure the predictive ability of the recommender systems is the All But One protocol [5] with 10-fold cross validation, i.e., 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 is used to build the recommendation model. For each user in the test set T_e, an item is hidden as a singleton set H. The remaining items represent the set of observable items O, that is used in the recommendation. Then, we compute Mean Average Precision ($MAP@N$), where N equals 5 and 10 recommendations. For each configuration and measure, the 10-fold values are summarized by using mean and standard deviation. To compare two recommendation algorithms, we applied the two-sided paired t-test with a 95% confidence level.

In our empirical evaluation, we used the 4 most similar items to make the recommendations and 0.1 as a threshold in *Filter PoF* to filter out the recommendations, since these values provided the best results for this experiment.

F. Results

In Table III, we show the results of our ranking evaluation by means of $MAP@N$. The results are obtained at four values of the combination factor ($\alpha = 0.3$, $\alpha = 0.5$, $\alpha = 0.7$ and $\alpha = 1$) and at three granularity levels, as described in Section IV-A. For each value of combination factor we also have the weights of each type of privileged information. To facilitate the understanding of the results, we mentioned the weight of technical information as BOW, the weight of the named entities as NE and the weight of domain terms as DT.

TABLE III: Comparing the context-aware recommendation algorithms against the IBCF algorithm. The values that are not statistically better than baseline (p-value>0.05) are in colored cells and the best results (higher values of MAP) for each algorithm are in boldface.

MAP results for $\alpha = 0.3$ ($BOW = 70\%$ - $EN = 10\%$ - $DT = 20\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3132490	0.3297517	0.1914374	0.3089337	0.3227017	0.3386906	0.1930766
$\{15, 20\}$	0.2991455	**0.4190486**	**0.4615741**	0.3435903	0.3089337	**0.4228415**	**0.4651130**	0.3446704
$\{2, 7\}$	0.2991455	0.3315996	0.3992454	**0.4267015**	0.3089337	0.3397160	0.4056149	**0.4278348**

MAP results for $\alpha = 0.3$ ($BOW = 70\%$ - $EN = 20\%$ - $DT = 10\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3281412	0.3464556	0.1823619	0.3089337	0.3382604	0.3569279	0.1842473
$\{15, 20\}$	0.2991455	**0.4242335**	**0.4496521**	0.3210554	0.3089337	**0.4289756**	**0.4537685**	0.3231635
$\{2, 7\}$	0.2991455	0.3220141	0.3754638	**0.4181398**	0.3089337	0.3298058	0.3815615	**0.4196089**

MAP results for $\alpha = 0.3$ ($BOW = 70\%$ - $EN = 15\%$ - $DT = 15\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3338906	0.3478617	0.1865615	0.3089337	0.3437439	0.3571797	0.1876844
$\{15, 20\}$	0.2991455	**0.3702801**	**0.4175135**	0.3040427	0.3089337	**0.3762570**	**0.4227551**	0.3053225
$\{2, 7\}$	0.2991455	0.3260990	0.3891643	**0.4063760**	0.3089337	0.3356745	0.3960491	**0.4077946**

MAP results for $\alpha = 0.5$ ($BOW = 50\%$ - $EN = 20\%$ - $DT = 30\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3482771	0.3792558	0.2252132	0.3089337	0.3583244	0.3886332	0.2267441
$\{15, 20\}$	0.2991455	**0.3895239**	**0.4189020**	0.2713655	0.3089337	**0.3952202**	**0.4239535**	0.2720121
$\{2, 7\}$	0.2991455	0.3409991	0.3962494	**0.4187704**	0.3089337	0.3504907	0.4038026	**0.4200874**

MAP results for $\alpha = 0.5$ ($BOW = 50\%$ - $EN = 30\%$ - $DT = 20\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3468419	0.3675248	0.1904683	0.3089337	0.3567991	0.3774468	0.1912013
$\{15, 20\}$	0.2991455	**0.3854856**	**0.4314550**	0.3205283	0.3089337	**0.3916562**	**0.4366166**	0.3208468
$\{2, 7\}$	0.2991455	0.3250967	0.3715466	**0.4021174**	0.3089337	0.3342467	0.3790192	**0.4037792**

MAP results for $\alpha = 0.5$ ($BOW = 50\%$ - $EN = 25\%$ - $DT = 25\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3228372	0.3427176	0.1882432	0.3089337	0.3318865	0.3514913	0.1896786
$\{15, 20\}$	0.2991455	**0.3742469**	**0.4267049**	0.3011386	0.3089337	**0.3795265**	**0.4310578**	0.3023229
$\{2, 7\}$	0.2991455	0.3399507	0.3897151	**0.4343196**	0.3089337	0.3497889	0.3975618	**0.4358158**

The presented results are for the three context-aware recommendation algorithms (*C. Reduction*, *Weight PoF* and *Filter PoF*), and also for the baseline IBCF. The *DaVI-BEST* results are not presented because they are equivalent to the IBCF results. So, there is no improvement by using this algorithm and the contextual information extracted with our proposal.

The analysis of the results can be divided into 3 questions: 1) What is the algorithm with the best results?; 2) What is the granularity with the best results?; and 3) What is the value of combination factor with the best results?

Answering the first question, we can observe that the algorithm *Weight PoF* presented the best results. This algorithm was also better than the baseline with statistical significance in all the experiments. For the second question, each algorithm was better at different granularity levels. The *C. Reduction* and *Weight PoF* algorithms presented the best results at configuration $\{15, 20\}$, while the *Filter PoF* algorithm presented the best results at configuration $\{2, 7\}$ (topics more specifics). The topics extracted by the configuration $\{50, 100\}$ presented values of MAP not as high as for the others configurations, which shows that it is better to consider more specific topics and in larger amount.

Analyzing the values of combination factor, to answer the third question, *C. Reduction* and *Weight PoF* presented the best results for combination factor $\alpha = 0.3$, ($BOW = 70\%$ - $EN = 20\%$ - $DT = 10\%$) and ($BOW = 70\%$, $EN = 10\%$, $DT = 20\%$) respectively, and *Filter PoF* for combination factor $\alpha = 0.5$ ($BOW = 50\%$ - $EN = 25\%$ - $DT = 25\%$).

TABLE III: (Continued) Comparing the context-aware recommendation algorithms against the IBCF algorithm. The values that are not statistically better than baseline (p-value>0.05) are in colored cells and the best results (higher values of MAP) for each algorithm are in boldface.

MAP results for $\alpha = 0.7$ ($BOW = 30\%$ - $EN = 20\%$ - $DT = 50\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3253358	0.3472713	0.2017587	0.3089337	0.3355101	0.3572727	0.2033553
$\{15, 20\}$	0.2991455	**0.3694954**	**0.4067977**	0.2962737	0.3089337	**0.3763551**	**0.4126285**	0.2969542
$\{2, 7\}$	0.2991455	0.3340217	0.3949698	**0.4194938**	0.3089337	0.3433043	0.4022932	**0.4207608**

MAP results for $\alpha = 0.7$ ($BOW = 30\%$ - $EN = 50\%$ - $DT = 20\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3364125	0.3512662	0.1996443	0.3089337	0.3445030	0.3599979	0.2008481
$\{15, 20\}$	0.2991455	**0.3606417**	**0.3957279**	0.2848784	0.3089337	**0.3668573**	**0.4012059**	0.2859318
$\{2, 7\}$	0.2991455	0.3303324	0.3840207	**0.4041100**	0.3089337	0.3393601	0.3918221	**0.4059656**

MAP results for $\alpha = 0.7$ ($BOW = 30\%$ - $EN = 35\%$ - $DT = 35\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3269617	0.3501964	0.2149991	0.3089337	0.3359110	0.3587213	0.2163066
$\{15, 20\}$	0.2991455	**0.3809494**	**0.4162618**	0.2924863	0.3089337	**0.3880923**	**0.4215766**	0.2940350
$\{2, 7\}$	0.2991455	0.33008407	0.3867889	**0.4177925**	0.3089337	0.3400565	0.3944094	**0.4188557**

MAP results for $\alpha = 1$ ($BOW = 0$ - $EN = 30\%$ - $DT = 70\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3045573	0.3232279	0.1299230	0.3089337	0.3128119	0.3314886	0.1300847
$\{15, 20\}$	0.2991455	**0.3447523**	**0.3808720**	0.3499418	0.3089337	**0.3538197**	**0.3893160**	0.3515182
$\{2, 7\}$	0.2991455	0.3151632	0.3676126	**0.4129423**	0.3089337	0.3255337	0.3755165	**0.4138143**

MAP results for $\alpha = 1$ ($BOW = 0\%$ - $EN = 70\%$ - $DT = 30\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3042942	0.3212988	0.1327955	0.3089337	0.3117196	0.3287964	0.1333247
$\{15, 20\}$	0.2991455	**0.3672521**	**0.4010373**	0.3059922	0.3089337	**0.3743193**	**0.4067024**	0.3074584
$\{2, 7\}$	0.2991455	0.3218124	0.3807921	**0.4232245**	0.3089337	0.3315916	0.3882532	**0.4245341**

MAP results for $\alpha = 1$ ($BOW = 0\%$ - $EN = 50\%$ - $DT = 50\%$)								
Granularities	$MAP@5$				$MAP@10$			
	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*	IBCF	*C. Reduction*	*Weight PoF*	*Filter PoF*
$\{50, 100\}$	0.2991455	0.3295935	0.3444711	0.1806840	0.3089337	0.3375443	0.3518203	0.1821174
$\{15, 20\}$	0.2991455	**0.3918618**	**0.4245879**	0.3416817	0.3089337	**0.4004201**	**0.4325431**	0.3437888
$\{2, 7\}$	0.2991455	0.3151337	0.3627965	**0.3907416**	0.3089337	0.3239142	0.3696845	**0.3918403**

In the graphic of Figure 2, we can analyze the best results of our experiments, i.e., the results for $\alpha = 0.3$ ($BOW = 70\%$, $EN = 10\%$ and $DT = 20\%$). The x-axis represents the granularities levels while the y-axis represents the values of MAP@10. Each line of the graphic is a recommender algorithm. It is evident that the three context-aware algorithms presented better results than the baseline *IBCF*, only *Filter PoF* presented a lower value of MAP at the granularity $\{50, 100\}$. At the granularity $\{2, 7\}$, this same algorithm presented better results than the others algorithms, what shows that this algorithm presents high values of MAP when topics more specifics are used. The algorithms *Weight PoF* and *Filter PoF* presented the best values of MAP at the granularity $\{15, 20\}$, being the *Weight PoF* the best of them.

V. Conclusion

In this paper, we proposed to use contextual information from topic hierarchies, constructed by LIHC method, to improve the accuracy of context-aware recommender systems. The topic hierarchies was constructed considering traditional bag-of-words (technical information), and the combination of named entities (privileged information I) and domains terms (privileged information II). The empirical evaluation showed that by using topics from the topic hierarchies with combined privileged information as contextual information, context-aware recommender systems can provide better recommendations. The contextual information obtained from the three topic hierarchies improved the recommendations in 3 out of 4 recommender systems evaluated in this paper: *C. Reduction*, *Weight PoF* and *Filter PoF* (in most of the experiments).

As future work, we will finish some experiments in which we are comparing the combined use of the two types of privileged information against the results of our previous studies using named entities and domain terms separately. Additionally, we will also compare our proposal against other baselines proposed in the literature.

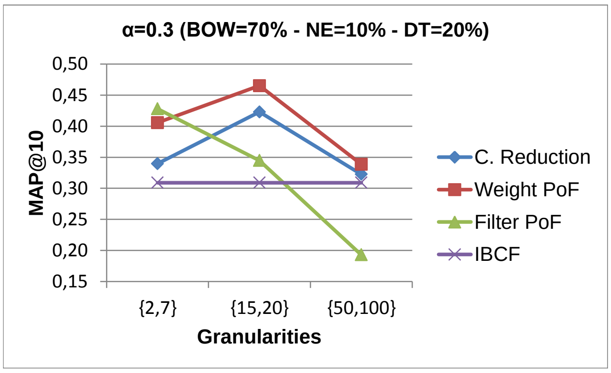

Fig. 2: Graphic with values of MAP@10 for $\alpha = 0.3$ ($BOW = 70\%$, $EN = 10\%$ and $DT = 20\%$).

Acknowledgment

The authors would like to thank FAPESP (grant #2010/20564-8, #2012/13830-9, and #2013/16039-3, São Paulo Research Foundation (FAPESP)) and CAPES/Brazil for financial support.

References

[1] S. Aciar. Mining context information from consumer's reviews. In *CARS'10: Proceedings of the 2nd Workshop on Context-Aware Recommender Systems*, 2010.

[2] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[3] G. Adomavicius and A. Tuzhilin. Context-Aware Recommender Systems. In *Recommender Systems Handbook*, pages 217–253. Springer, 2011.

[4] K. Bauman and A. Tuzhilin. Discovering Contextual Information from User Reviews for Recommendation Purposes. In *CBRecSys '14: Proceedings of Workshop on New Trends in Content-based Recommender Systems*, pages 2–9, 2014.

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[6] M. T. Cabré and R. E. J. Vivaldi. Automatic term detection: a reviw of current systems. In D. Bourigault, C. Jacquemin, and M.-C. L'Homme, editors, *Advances in Computational Terminology*, pages 53–88. John Benjamins, Amsterdam / Philadelphia, 2001.

[7] N. Cardoso. REMBRANDT - Reconhecimento de Entidades Mencionadas Baseado em Relaes e ANlise Detalhada do Texto. In *Desafios na avaliao conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, pages 195–211. Linguateca, 2008.

[8] M. S. Conrado. *Extração automática de termos simples baseada em aprendizado de máquina*. PhD thesis, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo - USP, 2014.

[9] M. S. Conrado, T. A. S. Pardo, and S. O. Rezende. 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*, pages 16–23, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[10] M. Deshpande and G. Karypis. Item-based top-N Recommendation Algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.

[11] A. K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

[12] M. A. Domingues, A. M. Jorge, and C. Soares. Dimensions As Virtual Items: Improving the Predictive Ability of top-N Recommender Systems. *Inf. Process. Manage.*, 49(3):698–720, 2013.

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

[14] N. Hariri, B. Mobasher, R. Burke, and Y. Zheng. Context-Aware Recommendation Based on Review Mining. In *Proc. of the 9th Workshop On Intelligent Techniques for Web Personalization and Recommender Systems*, pages 30–36, 2011.

[15] S.-S. Ho, M. Lieberman, P. Wang, and H. Samet. Mining Future Spatiotemporal Events and Their Sentiment from Online News Articles for Location-aware Recommendation System. In *Proc. of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 25–32, 2012.

[16] Y. Li, J. Nie, and Y. Zhang. Contextual Recommendation Based on Text Mining. In *Proc. of the 23rd International Conference on Computational Linguistics: Posters*, pages 692–700, 2010.

[17] R. Marcacini and S. O. Rezende. Incremental Hierarchical Text Clustering with Privileged Information. In *Proc. of the 2013 ACM Symposium on Document Engineering*, 2013.

[18] R. M. Marcacini and S. O. Rezende. Torch: a tool for building topic hierarchies from growing text collections. In *IX Workshop on Tools and Applications. In 8th Brazilian Symposium on Multimedia and the Web*, 2010.

[19] A. Mikheev, M. Moens, and C. Grover. Named Entity Recognition without Gazetteers. In *Proc. of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, pages 1–8, 1999.

[20] U. Panniello and M. Gorgoglione. Incorporating Context into Recommender Systems: An Empirical Comparison of Context-based Approaches. *Eletronic Commerce Research*, 12(1):1–30, 2012.

[21] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender System Handbook*. Springer, 2011.

[22] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1983.

[23] S. Sekine. Named Entity: History and Future. 2004.

[24] C. V. Sundermann, M. A. Domingues, R. M. Marcacini, and S. O. Rezende. Using topic hierarchies with privileged information to improve context-aware recommender systems. In *BRACIS '14: Proceedings of Brazilian Conference on Intelligent Systems*, pages 61–66, Oct 2014.

[25] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22:544 – 557, 2009.